Hunt like a Dragonfly and Strike like a Drone

Simulating games of pursuit and evasion to optimize quadcopter control for insect pest interception in greenhouses

Reinier Vos

TU Delft

Hunt like a Dragonfly and Strike like a Drone

Simulating games of pursuit and evasion to optimize quadcopter control for insect pest interception in greenhouses

Thesis report

by

Reinier Vos

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on October 4, 2024 at 10:00

Thesis committee:

Chair:	Dr. Ir. C. de Wagter
Supervisors:	Prof. Dr. G.C.H.E. de Croon
	Dr. M. Yedutenko
External examiner:	Dr. J.H. Boyle
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	November, 2023 - October, 2024
Student number:	4663160

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology

Preface

The completion of this thesis marks the end of my three-year-long journey towards the completion of two MSc. degrees in Quantitative Finance and Aerospace Engineering. While it is my broad interest and strategic mindset that has allowed me to formulate my ambitions, it is my relentless drive that has made me succeed at them. Through my studies, I have honed my practical skills and theoretical understanding with regard to applied statistics, control theory and system design. Over the span of this journey, I can clearly identify the growth I have experienced and am excited to see the new opportunities that lie ahead of me. I feel intense gratitude for my friends and family who have supported over these formative years and would like to thank my current thesis supervisors in guiding me during this project.

In this work, the aforementioned scientific fields are clearly represented to formulate my toughest challenge yet. The idea for this work was formulated after discussing advancements in biologically inspired artificial intelligence with Guido de Croon. After agreeing on the potential of this novel methodology, we sought out a challenging context to test the feasibility of our ideas. Together with Matthew Yedutenko and the team at PATS, we formulated an even more ambitious route; focussing on the application of powerful artificial intelligence to simulated conflicts between airborne pursuers and evaders. In this aim, we attempt to resemble evolution through dueling systems and, ultimately, design drone hunters capable of eliminating actual insect pests in green houses. At this final stage of research, I have accepted that our ambition transcends the work of a single MSc. thesis and can only say there is more work to be done. In all, I hope to one day see this or a similar solution applied in practice.

Thank you for reading this work.

- Reinier Vos, Rotterdam, September 2024

Contents

List of Figures vi

List of Tables viii

I Introduction 1

II Scientific Article 3

III Literature Review 23

1 Introduction 24

2 Strategies for pursuit and interception 25
2.1 Three strategies . . . 25
2.2 Motion camouflage . . . 29
2.3 Mixing MCPG to ensure interception . . . 36
2.4 Final remarks . . . 38

3 Games of pursuit and evasion 39
3.1 Differential games . . . 39
3.2 Deep Reinforcement Learning methods . . . 40
3.3 Insights from implementations . . . 44
3.4 Final remarks . . . 45

4 Neural pursuit controllers 46
4.1 Artificial neural networks . . . 46
4.2 Liquid gated synapses . . . 48
4.3 Liquid time-constant networks . . . 49
4.4 Neural Circuit Policies . . . 51
4.5 LTC-NCP implementations . . . 52
4.6 Final remarks . . . 54

5 Conclusion 54
5.1 Challenges and Opportunities . . . 54
5.2 Conclusion . . . 55
5.3 Future Research . . . 55

6 Literature review tables 57

IV Closure 59

References 67

Nomenclature

List of Abbreviations

A2C Synchronous Advantage Actor Critic

A3C Asynchronous Advantage Actor Critic

AI Artificial Intelligence

ANN Artificial Neural Network

BC(E) Behavioural cloning (Expert)

BPTT Back-Propagation Through Time

CATD Constant Absolute Target Direction

CCL Camouflage Constraint Line

CfC Closed-Form Continuous network

CL Curriculum Learning

CNN Convolutional Neural Network

CPN Custom Proportional Navigation

CT-RNN Continuous-Time recurrent network

CTDE Centralized Training with Decentralized Execution

DDPG Deep Deterministic Policy Gradient

DEC-POMDP Decentralized Partially-Observable Markov Decision Problem

DNN Deep Neural Network

DOF Degree of freedom

DP Deviated-Pursuit

DQN Deep Q network

DRL Deep Reinforcement Learning

EA Evolutionary Learning

EOM Equations of Motion

GRU Gated Recurrent Unit

HJI-PDE Hamilton-Jacobi-Isaac Partial Differential Equation

HoF Hall-of-Fame

IMU Inertial Measurement Unit

INDI Incremental Nonlinear Dynamic Inversion

INT Interception

KPSS Kwiatkowski–Phillips–Schmidt–Shin

LOS Line-of-Sight

LQR Linear Quadratic Regulator

LSTM Long Short-Term Memory

LTC Liquid Time-Constant network

M3DDPG Minimax Multi-agent Deep Deterministic Policy Gradient

MADDPG Multi-Agent Deep Deterministic Policy Gradient

MADRL Multi-Agent Deep Reinforcement Learning

MAPPO Multi-Agent Proximal Policy Gradient

MAV Micro-Air Vehicle

MC Motion Camouflage

MCPG Motion Camouflage Proportional Guidance

ML Machine Learning

MLP Multi-Layer Perceptron

MSE Mean Squared Error

NA Neural Activation

NCP Neural Circuit Policy

NODE Neural Ordinary Differential Equation

PCA Principal Components Analysis

PEG Pursuit-Evasion Game

PID Proportional-Integral-Derivative

PN Proportional Navigation

PP	Pure-Pursuit
PPO	Proximal Policy Gradient
RNN	Recurrent Neural Network
RQ	Research Question
SA	Synaptic Activation
SADRL	Single-Agent Deep Reinforcement Learning
TRUNC	Truncation

List of Symbols

β	Model parameters
$\boldsymbol{\lambda}$	Line-of-Sight angle
$\boldsymbol{\Omega}$	Body angular rates (pitch, roll, yaw rate)
$\boldsymbol{\Theta}$	Euler angles
$\boldsymbol{d}$	distance
$\boldsymbol{s}$	Game state
$\boldsymbol{x}$	Agent state
Γ	Range vector correlation
γ	Reward discount factor
a	Acceleration
$\mathbf{C_d}$	Drag coefficient
F	Resultant force
p	Position
v	Velocity
$\mathcal{A}$	Action space
$\mathcal{N}$	Normal distribution
$\mathcal{O}$	Observation space
$\mathcal{P}$	State transition probability space
$\mathcal{U}$	Uniform distribution
μ	Distribution mean
π	Action policy
σ / Σ	Distribution variance
R	Rotation matrix
r	Reward
T	Thrust
t	Time
u	Pursuer input signal
v	Evader input signal

List of Figures

2.1 Examples of pursuit strategy trajectories for an evader exhibiting random motion in two dimensions. Classical or pure pursuit (a), constant absolute target direction or CATD (b), deviated pursuit/constant bearing (c). Notice that deviated pursuit strategy slightly rotates the LoS based to retain a consistent relative heading of 0.3 rad, while CATD's retains a strictly parallel line-of-sight. Image retrieved from [13]. 26

2.2 Particle reference frames with tangential (**x**) and normal (**y**) velocities for pursuer (p) and evader (e), with relative velocity scalar ν. Image retrieved from [13]. 26

2.3 Geometric representations of the three pursuit manifolds related to cost variables $\Lambda(\theta)$ and Γ. Cases denote manifolds for (a) pure pursuit, (b) deviated or constant bearing pursuit and (c) motion camouflage pursuit. Image retrieved form [13]. 27

2.5 Sustained straight evader trajectory and pursuer trajectory with consistently parallel CCL, implying equivalence between constant bearing and CATD pursuit strategies. Image retrieved from [8]. 29

2.4 Variants of motion camouflage achieved by the pursuer from the perspective of the evader for different focal point scenarios. Where the first two cases denote a finite focal point, the last case (C) illustrates a focal point at infinity. Image retrieved from [21]. 30

2.6 Phase plots describing the dynamics of the strategy population probability through a simulated evolutionary game. Notice how the populations start from diverse initial positions, yet consistently converge to the CATD strategy (bottom right vertex), indicating strategy superiority. (a-b) visualize two representative outcomes from simulations with various (stochastic, linear & circular) evader trajectories over 18 trials. (c-d) visualize two representative outcomes from simulations with stochastic evader trajectories over 75 trials. (e-f) visualize two representative outcomes from simulations with circular evader trajectories with random turning rates over 50 trials. Image retrieved from [13]. 31

2.7 Srinivasan et al.'s perspective on finite focal point motion camouflage, where black and white dots indicate positions of the pursuer and evader, respectively. (a) Definitions of fixed point distance (ρ), pursuer's required lateral displacement ($\Delta\lambda$) and evader relative rotation ($\Delta\theta$). (b) Lateral displacement is dependent on the range to the evader. Image retrieved from [29]. 32

2.8 Anderson et al.'s controller and visual input overview. (A) The controller consists of three interconnected recurrent subsystems computing the distance to finite focal point (*Dist*) as well as the direction (*Dir*) and rotation (*Rot*) commands). (B) Input azimuth and elevation angles defined as θ and ϕ respectively. Image retrieved from [30]. 32

2.9 Rano's controller input definitions for two-dimensional motion camouflage with finite focal point. $\mathbf{r}_s$, $\mathbf{r}_t$ and **F** indicate the position of the pursuer (*shadower*, subscript s), evader (*target*, subscript t)) and focal point respectively and **v** indicates the velocity. The relative direction, the relative evader angle and the instantaneous angle of the CCL are indicated by d, β, and α respectively. Image retrieved from [21]. 33

2.10 Rano's reward terms utilized for controller identification, displayed in terms of reference frames and reward magnitude. Image retrieved from [21]. 34

2.11 Illustration of the (slight) rotation of the finite focal point. Blue and red indicate pursuer and evader trajectory, respectively. It is hypothesized how this small and negligible non-stationarity of the finite focal point becomes imperceivable from the perspective of the evader conditional on it being subject to some angular slack condition, ϵ_m. 34

2.12 Illustration defining the hypothesized relationship between effective gain setting μ' in the CATD control law (Equation 2.6) and pure pursuit (left) and finite (center) and infinite (right) motion camouflage trajectories respectively. Blue and red indicate pursuer and evader trajectory, respectively. The hypothesis defines that for unity gain of μ' in the CATD controller, pure pursuit is implemented with clearly perceivable rotation of the CCL (i.e. no motion camouflage). For higher gains, the rotation of the CCL is hypothesized to reduce. This ultimately results in finite focal point motion camouflage (center), assuming that the small rotational rate of the focal point is imperceivable from the perspective of the evader (see close up in Figure 2.11). For even larger gain settings, the CCL becomes truly stationary and infinite focal point motion camouflage (right) is achieved. 36
2.13 Sequential overview of replicated dragonfly control strategy through attempted cancellation of *prey-image slippage* with respect to the fixation spot/fovea by Plunkett and Chance[35][36]. Image retrieved from `https://www.osti.gov/servlets/purl/1894019`. 37
2.14 Empirical trajectory Γ (CATD objective, *range vector correlation*) as a function of *time to capture* for dragonfly hunting fruitfly (f) and artificial prey (g) in experimental setting respectively. Image retrieved from [37]. Recall that *parallel navigation* refers to the case where the CATD strategy (Equation 2.6) achieves optimal trajectory, i.e. whenever $\Gamma = -1$ (Equation 2.2). 38

3.1 Multi-agent scenario with centralized training and decentralized execution principle in actor-critic (π_i & Q_i respectively) system setup. o_i indicates the (partial) game state observed by an individual agent, while a_i indicates the agent's action. Image retrieved for MADDPG architecture from [77]. 42

4.1 Deep and residual neural networks connection schematic. Image retrieved from [107]. . . . 47
4.2 Possible evaluation points for discrete (left) and continuous (right) transformations as implied by residual/recurrent and ODE network types. *Residual neural network* (ResNet)[118] describes an recurrent connection equal to the identity. Image retrieved from [117]. 47
4.3 Electrical representation of non-spiking neuron. Note that this representation defines the neuron-synapse connection as autonomous, void of external inputs (u). Image retrieved from [107]. 48
4.4 Visualization of complexity scope as measured through latent-spaced trajectory length (with dimension reduced through PCA). The initial trajectory is a circular motion, which is sequentially transformed into a more complex pattern through the layers. Notice how the latent trajectory becomes increasingly complex as it passes through the layers. Image retrieved from [121]. 50
4.5 Closed-form continuous depth (CfC) network architecture according to the definition in Equation 4.8. Image retrieved from [124] 52
4.6 Visualized neural circuit policy design algorithm for undefined neuron type and layer size. Image retrieved from [105]. 52
4.7 Interpretation of single neuron (of decoupled network) control mean response in decision tree format subject to certain local heading error (μ) and lateral deviation (d) states for NCP-LTC controller applied to lane keeping task. An interpretable decision tree follows from algorithm methodology by Wang et al. [128]. Image retrieved from [128]. 54

List of Tables

6.1 Pursuit-evasion implementations with a single optimized agent. The *setup* contains a code describing the number of dimensions (D), as well as the number of pursuer (P/p) and evader (E/e) agents, where N indicates multiple > 1. A capital letter (e.g. E), as opposed to a lower case one (e.g. e), indicates a data-driving technique is used to identify the controller for the respective agent. Additional abbreviations for models and algorithms have been introduced in Chapter 3 and Chapter 4. The order of sources in these tables reflects that of the order of discussion in these chapters. 57

6.2 Pursuit-evasion implementations with multiple optimized agents. The *setup* contains a code describing the number of dimensions (D), as well as the number of pursuer (P/p) and evader (E/e) agents. A capital letter (e.g. E), as opposed to a lower case one (e.g. e), indicates a data-driving technique is used to identify the controller for the respective agent. Additional abbreviations for models and algorithms have been introduced in Chapter 3 and Chapter 4. The order of sources in these tables reflects that of the order of discussion in these chapters. 58

Part I

Introduction

Micro-air vehicles (MAVs) have seen a significant rise in prominence over the past few decades, finding diverse applications across industries, including agriculture. These small, autonomous drones are already being used to enhance crop yields through detailed monitoring, providing farmers with real-time data on crop health. Furthermore, this detailed monitoring has allowed farmers to accurately quantify the effect of certain factors such as insect pests on the yield of their crop. However, while MAVs can assist with awareness, their direct usage as a solution to combat insect pests is limited and is traditionally controlled using insecticides. An alternative to such chemicals is currently developed by PATS[1], a Delft-based agro-tech startup that develops autonomous drone systems for pest monitoring and control in greenhouses. Specifically, their ambition is to equip MAVs with autonomous capabilities to intercept and eliminate these pests, offering a new frontier in pest management that is both more sustainable and technologically advanced.

Inspiration for designing MAV controllers capable of such tasks can be drawn from nature, where airborne predators like dragonflies and bats continuously evolve in a biological arms race with their prey. Expert hunters such as the dragonfly demonstrate sophisticated pursuit strategies, achieving interception success rates as high as 80%[2]. By studying how these hunters efficiently track and capture their prey, researchers can extract principles that can be applied to the development of robust and competitive pursuit controllers for practical domains concerned with target interception. With regard to the aforementioned use case in greenhouses, such bio-inspired strategies directly hold the potential to enhance the precision and interception efficacy of insect pests by MAVs as designers attempt to mimic or even surpass the effectiveness of natural hunters.

The intent of this research is to contribute to this ambition through the development and assessment of an optimization framework to identify robust controllers for autonomous MAVs that can consistently intercept insect pests. To this end, this report contains a comprehensive review of literature focussing on natural predator behavior, optimization in game theory and advancements in neural nonlinear controllers. Using a differential game framework, this study subsequently models a pursuit-evasion simulation scenario between a drone pursuer and an insect evader that are asymmetric to each other in terms of characteristics, capabilities and objectives. Through multi-agent deep reinforcement learning (i.e. optimization of both pursuer and evader), the controllers for both agents are optimized in an attempt to identify robust pursuit strategies which are further assessed on resemblance to natural behavior. In order to focus on a practical solution, the study imposes realistic sets of constraints and limitations on the agent's vehicle dynamics and observations sets. Ultimately, the research aims to explore the benefits of multi-agent optimization and nonlinear parameterized controllers. Specifically, it aims to assess whether they can outperform traditional benchmark control laws for interception as well as parameterized controllers that are identified through the single-agent alternative in intercepting maneuverable insect-like evaders.

Research Formulation

The ambition of this study is formulated into the following objective,

Research Objective

To develop a nonlinear parameterized pursuit controllers for autonomous quadcopters capable of intercepting insect-like evaders through multi-agent deep reinforcement learning applied to simulated games of pursuit and evasion

The objective is split into two separate research questions addressed in the scientific article and defined by,

Research Question 1

Do parameterized nonlinear controllers for quadcopter pursuers identified through multi-agent deep reinforcement learning outperform the single-agent alternative in terms of the interception rate of insect-like evaders?

and,

Research Question 2

Does natural predator behavior emerge from the co-evolution between a quadcopter pursuer and insect-like evader in simulated games of pursuit and evasion?

The subsequent parts in this report will attempt to provide an overview of all background information and previous literature to formulate a research strategy to obtain answers to the aforementioned questions. The scientific article will implement this research strategy and formulate conclusions based on the discovered results.

Structure of the Report

This thesis report consists of three parts. First of all, Part 2 describes the scientific article which has been set up to address the research question. Secondly, Part 3 describes a literature review which considers all the previous research and background information associated with the main scientific article in Part 2. Finally, Part 4 describes the final conclusions and recommendations which have been formulated as a result of the conducted research. The contents of the specific parts is specified further as follows.

The scientific article in Part 2 addresses the design of controllers capable of interception through the games of pursuit and evasion. The article is organized into six main sections, providing a flow from introduction to the conclusions and recommendations. The core of the article lies in its extensive Methodology in Chapter 2, which meticulously outlines the study's approach, including game definition, agent dynamics, observations, optimization techniques, and policy implementations. This is followed by a description of the experimental setup in Chapter 3, setting the stage for the Results and Discussion in Chapter 4 before drawing up recommendations in Chapter 5 and a final conclusion in Chapter 6.

The literature review in Part 3 addresses the comprehensive overview of previous literature relating to the scientific article. The article focusing on three main angles, namely pursuit and interception strategies in Chapter 2, definitions of differential games of pursuit and evasion in Chapter 3, as well as advances in and applications of neural controllers in Chapter 4. The article concludes with a comprehensive conclusion in Section 5.2; addressing challenges, opportunities, and future research directions.

Part II

Scientific Article

1 Introduction **4**

2 Methodology **6**

2.1 Game definition 6

2.2 Agent dynamics. 6

2.3 Observations 7

2.4 Optimization 9

2.5 Policies 13

3 Experimental setup **14**

4 Results & Discussion **16**

4.1 Game statistics 16

4.2 Online Confrontations 17

4.3 Offline Opogona recordings 19

4.4 Trial progression 21

5 Limitations & Future Research **23**

6 Conclusion **23**

Hunt like a Dragonfly and Strike like a Drone: Optimizing quadcopter control for insect pest interception through multi-agent deep reinforcement learning

R.W. Vos, Dr. M. Yedutenko, Prof. Dr. G.C.H.E. de Croon

***Abstract*—**

Insect pest elimination through MAV interception can reduce the need for insecticides and can contribute to sustainable agriculture. In this research, we analyze the feasibility of such solutions through simulated two-player differential games of pursuit and evasion with agents operating on minimalistic sets of biologically-plausible observations and optimized to control constrained vehicle models through deep multi-agent reinforcement learning. Our pursuer and evader agent, representing the quadcopter drone and insect pest respectively, are asymmetric in design, capabilities and objectives. Our results show that our quadcopter pursuer is consistently able to pursue and intercept a reactive insect-inspired evader as well as recordings of actual insect targets, achieving interception rates of 55% and 94% on these respective tasks. In comparison, pursuers alternatively optimized against non-reactive evaders or reactive drone-like evaders with symmetric capabilities, achieve an interception rate of only 42% for the same insect target recordings. Despite these promising results, we conclude that further research is needed to formally establish the superiority of multi-agent optimization in this asymmetric game scenario. Finally, we determine how emergent behavior and strategies resemble nature. During the confrontations, we observe that our pursuer mainly implements pure-pursuit as well as motion camouflage t o s ome degree; drawing comparison to the hunting strategy of dragonfly.

***Index Terms*—Differential games of pursuit and evasion, Simulated Evolution, Multi-agent Deep Reinforcement Learning, Credit Assignment Problem, Proportional Navigation, Motion Camouflage, N onlinear q uadcopter c ontrol, I nsect dynamics.**

I. Introduction

Dragonflies a re e xpert a irborne p redators c apable o f sophisticated pursuit with consistently high interception rates, upwards of 80%[1]. Through evolution, they have acquired heuristics to select assailable prey [2] and have developed a pursuit strategy that minimizes time-to-intercept, energy spent as well as perceivable visual cues[3]. Hence, the dragonfly a nd other natural predators have often served as inspiration for research and development of controllers capable of robust interception in fields f aced w ith s imilar o bjectives, s uch a s autonomous guidance of missiles[4] and drones[5]. This research aligns with this paradigm and draws inspiration from nature to design pursuit controllers for Micro-Air Vehicles (MAVs) capable of eliminating insect pests in greenhouses to reduce the need for insecticide and provide a more sustainable form of agriculture.

All three researchers are with Delft University of Technology, Faculty of Aerospace Engineering, Department of Control & Simulation, Micro-Air-Vehicle Laboratory (MAVLab). This scientific article is part of the thesis report for fulfillment of a MSc. in Aerospace by Reinier Vos defended on October 4, 2024

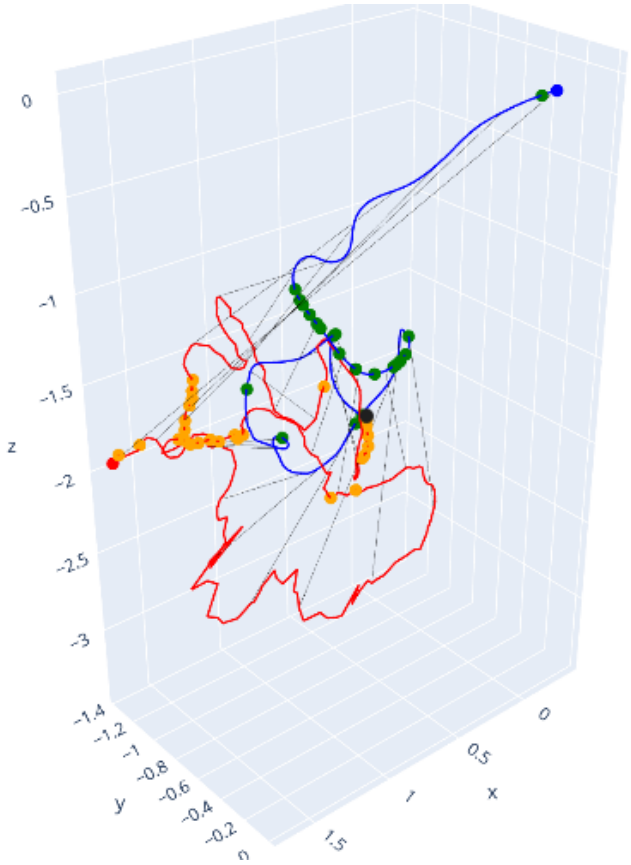


Fig. 1. Representative 3D visualization of insect target recording (red) intercepted (black dot) by our quadcopter pursuer (blue) implementing its strategy optimized through multi-agent reinforcement learning for games of pursuit and evasion. Orange and green dots indicate the evader or pursuer subject to a motion-camouflaged adversary at specific time steps, respectively. Complete set of trajectory visualizations available at https://github.com/rwvosTUD/DragonfliesAndDrones.git.

While robust control laws exist for the capture of targets moving along predictable trajectories, intercepting reactive

evaders evolved through natural arms-races is no small feat. For this class of targets, higher-order kinematic quantities (acceleration, jerk, etc.) can become completely unreliable, due to unknown target dynamics and (frequency) limits of sensory equipment (noise and update frequency). Moreover, last-moment escape strategies and general agility of evaders means that success of pursuit is hard to predict until the moment of interception itself, as observed in moths hunted by bats[6].

In recognition of these complexities, differential game theory is often employed as a framework in order to derive robust controllers in consideration of both agents' asymmetric capabilities and adversarial objectives. Although analytical solutions can exist for simplified formulations of these *games of pursuit and evasion* [7], in other cases (approximations of) optimal strategies can be obtained through deep reinforcement learning (DRL)[8][9].

For pursuit-evasion games, implementations of DRL often only consider optimization of the pursuit controller, instead of the evader's as well [10][11][12][13][13]. These works identify successful controllers, yet rely on the aforementioned higher-order kinematic estimates, stemming from known target dynamics, and exclude sudden evasive maneuver capabilities. In opposition, Gaudet et al.[14] attain a competitive strategy implementation by optimizing a pursuit controller to capture missile evaders following a set of pre-defined escape strategies, using only the true line-of-sight angles and angular rates to the target as an observation set. In practice, these lower-order kinematic quantities can be estimated more accurately than higher-order ones. Hence, these works emphasize how inspiration from nature can lead to reliable controllers operating on minimalistic observations without the need to sacrifice performance.

In works that align with the differential game definition, optimization of both agents in this pursuit evasion scenario is shown to improve the pursuit controller's robustness to adversarial behavior [15][16][17][18][19]. Moreover,these works observe some similarities to biological behavior, with regard to the utilization of asymmetric capabilities. For instance, evaders elongate/escape pursuit by taking advantage of their improved agility observed through tighter turns or last-moment trajectory adjustments into the bounds of the pursuer's turning radius. While these works highlight that natural pursuit and evasion behavior can arise in simulation through multi-agent optimization, they lack the dedicated and representative vehicle models required to formulate and assess the feasibility of autonomous artificial hunters in eliminating insect pests. In addition, this assessment is further limited as their analyses remain subject to various simplifications including two-dimensional game scenarios, ideal and (near) complete state observation without noise and/or delay, as well as the use of agent vehicles with immediate responses. Hence, it is generally undetermined how the proposed multi-agent optimization methods hold up in a more realistic and restrictive version of the considered game scenarios and to what extent emergent strategies resemble natural behavior under these conditions.

Therefore, in this research we attempt to identify a parameterized controller subject to a set of imperfect and biologically-plausible observations subject to noise and delay, meant for onboard use in an autonomous MAV, capable of consistent interception of insect pests. To this end, we implement a differential game of pursuit and evasion, with controllers optimized through multi-agent deep reinforcement learning. Our pursuer and evader agent, representing the drone and insect pest respectively, control non-ideal vehicle models that are asymmetric in design, capabilities and objectives. Our research intends to analyze the usefulness of multi-agent optimization as well as the benefit of parameterized nonlinear controllers over classical benchmark control laws for interception (e.g. proportional navigation) in this context of pursuit-evasion games between maneuverable insect evaders and faster drone pursuers. In addition, we assess whether pursuer behavior is affected by implicitly encouraging motion camouflage strategies and determine whether pursuer behavior can become more robust to adversarial strategies by implementing evolutionary lead and lag between agents.

This research is structured as follows. First, section II provides the methodology. Afterward, section III addresses the evaluation procedure. Then, the results are analyzed in section IV, followed by recommendations in section V and a conclusion in section VI.

II. Methodology

A. Game definition

In this research, the dynamical conflict between the hunter and its target is categorized as a differential game of pursuit and evasion, wherein a differential equation describes the agents' influence on the game's state and its outcome[20]. Formally, it is described as,

$$\dot{\mathbf{s}} = f\left(t, \mathbf{s}(t), u(t), v(t)\right) \tag{1}$$

where $\mathbf{s}(t)$ defines the *game state* and $u(t) = \mu(t, \mathbf{s}(t))$ and $e(t) = \nu(t, \mathbf{s}(t))$ provide a general formulation of the pursuer's and evader's feedback control policies under perfect observation of the complete state respectively. In our research, we intend to estimate the optimal feedback policies under partially observed and discretized approximation of the game state using deep reinforcement learning.

The finite multi-agent pursuit-evasion game is modeled as a *Decentralized Partially-Observable Markov Decision Problem* (DEC-POMDP)[21]. It is defined by the variables $\{\mathcal{S}, \mathcal{A}_p, \mathcal{A}_e, \mathcal{O}_p, \mathcal{O}_e, \mathcal{P}, r, \gamma\}$. For a given game state ($s_t \in \mathcal{S}$), agent i implements a continuous action $a_{i,t} \in \mathcal{A}_i$ conditional on its observation ($o_{i,t} \in O_i$) of said state, according to its stochastic policy $\pi_i(a_{i,t}|o_{i,t})$. Given both the agents states, the game transitions to state $s_{i,t+1}$ according to $\mathcal{P}_{s_t,s_{t+1}}^{a_{p,t},a_{e,t}}$, resulting in agent rewards, $\{r_{p,t}, r_{e,t}\}$. In our setting, adversarial behavior arises due to the conflicting definition of these rewards

designed to promote interception and escape respectively (to be discussed shortly). This behavior is learned through the optimization of parameter sets (β_i) of the stochastic policies' (π_i) and attempts to maximize the expected sum of rewards over the finite horizon, i.e. $\text{argmax}_{\beta_i} \sum_{t=0}^{T} \gamma^t r_{t,i}$. Episodes take place in a boundless arena and run until truncation ($TRUNC$) due to exceeding the time limit of 10 seconds or until termination due to interception (INT) i.e. in case of $d_t < d_{INT}$ with d_{INT} describing the interception distance set at 5 cm. Finally, note that practical description of the agent's state, action and observation vectors correspond to $\mathbf{x}_i(t)$, $\mathbf{u}_i(t)$ and $\mathbf{o}_i(t)$ provided in Equation 2 and Equation 4 respectively. These continuous definitions are determined at the control frequency/time step, given by their discretized representations: $\mathbf{x}_{i,t}$, $\mathbf{u}_{i,t}$ and $\mathbf{o}_{i,t}$.

B. Agent dynamics

We model our game setup by opting for identical equations of motion and command structure for both agents, albeit with different specifications and capabilities. Specifically, we opt for the quad-copter control system as utilized by Ferede et al.[22] For this system, agents control body angular rates ($\mathbf{\Omega} = \{p, q, r\}$) as well as thrust ($T$) with state and control inputs for agent i defined as,

$$\begin{aligned} \mathbf{x}_i &= [\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i, \mathbf{\Theta}_i, \mathbf{\Omega}_i, T_i]^T \in \mathbb{R}^{16}, \\ \mathbf{u}_i &= [\mathbf{\Omega}_{cmd,i}, T_{cmd,i}]^T \in \mathbb{R}^4, \end{aligned} \tag{2}$$

where $\mathbf{p}_i$, $\mathbf{v}_i$, $\mathbf{a}_i$ and $\mathbf{\Theta}_i$ indicates the agent position, velocity, acceleration, and attitude (through Euler angles i.e. $\{\phi, \theta, \psi\}$) respectively. Inline with Ferede et al.[22] command incorporation is proposed through a lower-level INDI controller[23], effectively described through a slower outer and faster inner cycle. Where the outer command cycle (subscript cmd) operates at a frequency of 100 Hz, the inner control INDI loop can be represented as a first-order delay model and represents the lower-level controller's performant and nonlinear implementation of the commanded references[23]. Hence, the associated equations of motion are defined as,

$$\begin{aligned} \dot{\mathbf{p}}_i &= \mathbf{v}_i & \dot{\mathbf{v}}_i &= -g_i \mathbf{e_3} + R(\mathbf{\Theta}_i)\mathbf{F}_{B,i} \\ \dot{\Theta}_i &= R'(\mathbf{\Theta}_i)\mathbf{\Omega}_i & \dot{\mathbf{\Omega}}_i &= (\mathbf{\Omega}_{cmd,i} - \mathbf{\Omega}_i)/\tau_{\Omega_i} \\ \dot{T}_i &= (T_{cmd,i} - T_i)/\tau_{T_i} \\ \text{with} \quad \mathbf{F}_{B,i} &= -(\mathbf{I_3 C_{d_i}})(R^T(\mathbf{\Theta}_i)\mathbf{v}_i) + T_i \mathbf{e_3}, \end{aligned} \tag{3}$$

where R is the rotation matrix from body to inertial frame and g is the gravitational acceleration. F_B represents the mass-normalized resultant force (i.e. acceleration) resulting from both (linear) velocity drag (coefficient vector $\mathbf{C}_d$) and thrust in the body reference frame. The relevant characteristics for both agents with regard to these dynamics are provided in Table I. Note that acceleration is not an explicit output of the integration scheme and post-pended using the equality ($\dot{\mathbf{v}} = \mathbf{a}$). Finally, note that in order to allow for proper exploration and avoid potential issues such as gimbal lock, integration involves the conversion of Euler angles to quaternions. Integration is performed using forward Euler discretization at 500 Hz (i.e. 5 times larger than the 100 Hz command frequency), where commands are considered of zero-order hold nature during the intervals.

	$Pursuer$	$Evader$
$T_{cmd,i}$	$[0, 2g]$	$[-5g, 5g]$
τ_{T_i}	0.03	0.01
$\{p, q, r\}_{cmd,i}$ $(= \mathbf{\Omega}_{cmd,i})$	$[-10\pi, 10\pi]$	$[-20\pi, 20\pi]$
τ_{Ω_i}	0.03	0.01
$\mathbf{C}_{d_i}$	$[0.5, 0.5, 0.5]^T$	$[\frac{1}{0.02}, \frac{1}{0.02}, \frac{1}{0.06}]^T$
g_i	g	0
τ_{perc_i}	0.03	0
τ_{inert_i}	0	0
$\mu_{\mathbf{x}_i}$	$[0, 0, 0]^T$	$[0, 0, 0]^T$
$\boldsymbol{\sigma}_{\mathbf{x}_i}$ $(\mathbf{I}_3 \boldsymbol{\sigma}_{\mathbf{x}_i} = \Sigma_{\mathbf{x}_i})$	$[0.002, 0.002, 0.002]^T$	$[0.002, 0.002, 0.002]^T$

TABLE I
CHARACTERISTICS FOR AGENT DYNAMICS AND AGENT OBSERVATIONS DESCRIBED IN EQUATION 3 AND EQUATION 4 RESPECTIVELY. g DESCRIBES THE GRAVITATIONAL CONSTANT ($9.81 m/s^2$). DRAG COEFFICIENTS FOR THE EVADER ARE EXPRESSED THROUGH THE INVERSE OF TIME SPAN TO EFFECTIVELY NULLIFY ANY REMAINING VELOCITY (I.E. $\tau^{-1}_{\mathbf{v}_{NULL}}$), WHICH ARE CONSIDERED MORE INTUITIVE/INFORMATIVE THAN THEIR CRUDE VALUES.

The two agents differ substantially in their characteristics, as can be seen in Table I. The pursuer represents a quadcopter drone with representative values for drag coefficients and thrust capabilities[22][24][25]. On the other hand, the evader possesses fast and large thrusting capability, yet its resultant force along the body z-axis is rapidly capped by drag forces of similar size. In addition, any *residual* velocity in the body x and y axis is promptly nullified. To summarize, The evader is capable of reaching maximum a speed of $3m/s$, yet generally operates at around $1m/s$ if maneuvering is taken into account. Finally, note that the evader is not subject to gravity, because the focus of this research is on robust pursuit controller design and its intent is not to design a stable insect evader model. Equally important is the fact that this choice generally stabilizes the z-coordinate of the evader as well as that of the pursuer, since the latter chases the former. Hence, the game space is virtually bounded in its z-coordinate, avoiding the need for additional measures to counteract crashing/sustained diving behavior through means such as termination conditions and scoring penalties.

Qualitatively, this system definition for the evader exhibits strong control of the displacement vector, resulting in a highly maneuverable agent capable of rapid smaller adjustments. On the other hand, it is not superior in terms of top speed, restricting the evader's potential in outrunning the pursuer. Hence, this definition is deemed a qualitative description rather than a quantitative representation of insect dynamics. Compared to the evader, the pursuer exhibits a superior top speed, yet with the disadvantage of more sluggish control incorporation. All in all, these differences in agent characteristics have been intentionally selected to qualitatively reflect the realistic asymmetry in capabilities one can observe between drone pursuers and insect evaders.

Initialization
Agents are initialized in an asymmetric manner. The pursuer always starts at the origin and the evader at a random point on the top half of a sphere with radius r_0 (i.e. positive z-coordinate). The initial control states $\{p_{i,0}, q_{i,0}, r_{i,0}, T_{i,0}\}$ are sampled from a uniform distribution with bounds at 25% and 75% of the limits reported in Table I. Furthermore, the pursuer's initial normalized velocity vector is normally distributed at the evader's normalized position vector ($\bar{\mathbf{v}}_{p,0} = \mathcal{N}(\bar{\mathbf{p}}_{e,0}, \mathbf{I}_3 \cdot \frac{1}{10})$). In contrast, the evader's normalized velocity vector is initialized on the unit sphere, yet with an orientation neither directly towards nor away from the pursuer. It is implemented by defining the unit sphere with its polar axis on the range vector and prohibiting sampling near the poles, i.e. inside the polar/zenith angular range $[\frac{1}{4}\pi, \frac{3}{4}\pi]$ and with any azimuthal angle. This type of initialization is done to avoid immediate head-on collisions or tail chases, respectively; which might hinder optimization due to the agents receiving reward for arbitrary actions at these game states. Both agents start with a velocity magnitude ($||\mathbf{v}_{i,0}||$) uniformly sampled from $\mathcal{U}(\frac{1}{2}; \frac{3}{2})$.

Furthermore, for both agents the corresponding attitude vector ($\mathbf{\Theta}_{i,0}$) is found that aligns the unit velocity vector in the inertial reference frame ($\bar{\mathbf{v}}_{i,0}$) with the fixed thrust orientation in the body reference frame (i.e. unit vector $\bar{F}_{B,T} = \mathbf{e}_3 = [0, 0, 1]$). For the latter step, we employ the *Kabsch algorithm* using SciPy software[26] to solve this *pointing problem* and find the corresponding attitude vector ($\mathbf{\Theta}_{i,0}$). Using this attitude and the definition of $\dot{\mathbf{v}}_i$ in Equation 3, we compute the initial acceleration vector, $\mathbf{a}_{i,0}$.

C. Observations

In this research, we focus on a biologically plausible and minimalistic set of observations from the perspective of the body reference frame, identical for both agents. Given the heavy reliance of predator insects such as dragonfly and killer flies on vision[27][28], they cannot rely on full state information of their prey. In particular, target range and information on higher order kinematic quantities (e.g. acceleration, jerk, etc.) are subject to considerable noise and uncertainties and unlikely to be used. In line with this motivation as well as results from prior studies[14], we limit these types of information by exclusively encoding target information through the line-of-sight (LoS) angles and angular rates. On the other hand, our study is limited as these inputs are assumed always available instead of using a restricted field-of-view. All in all, this approach implies a divergence from conventional theory on missile guidance similar to Gaudet et al.[14], whose controllers often rely on those higher-order kinematic quantities of the target, such as target acceleration in the *augmented proportional navigation* missile guidance law[4]. In order to maintain a stable system, we do provide information on the agent's own acceleration and attitude, deemed accessible through internal sensors such as IMU and/or gyroscopes and cleaned through the use of Kalman filters. Finally, the previous command/history of commands is provided to improve training/performance, following their use in successful practical applications[29][30]. All in all, the observation vector for agent i at time step t is defined as,

$$\begin{aligned}\tilde{\mathbf{o}}_i(t) =& \mathrm{O}\left(\tilde{\mathbf{x}}_{B_i,i}(t), \tilde{\mathbf{x}}_{B_i,j}(t), \mathbf{u}_i(t); \delta t, \tau_{perc,i}, \tau_{inert,i}\right) \\ =& [\tilde{\boldsymbol{\lambda}}_i(t-\tau_{perc,i}), \tilde{\dot{\boldsymbol{\lambda}}}_i(t-\tau_{perc,i}), \\ & \tilde{\mathbf{a}}_i(t-\tau_{inert,i}), \tilde{\mathbf{\Theta}}_i(t-\tau_{inert,i}), \\ & \mathbf{\Omega}_{cmd,i}(t-\delta t), T_{cmd,i}(t-\delta t)] \in \mathbb{R}^{17}\end{aligned}$$

with,

$$\begin{aligned}\tilde{\mathbf{x}}_{B_i,i}(t) =& R(\mathbf{\Theta}_i(t))^T \mathbf{x}_i(t) + \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}). \\ \tilde{\mathbf{x}}_{B_i,j}(t) =& R(\mathbf{\Theta}_i(t))^T \mathbf{x}_j(t) + \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}).\end{aligned} \tag{4}$$

Here, $\mathbf{x}_{B_i,i}(t)$ denotes the agent i's state in body reference frame of agent i (B_i) at time step t, where the rotation matrix is not applied to the non-Cartesian states i.e. $\{\mathbf{\Theta}, \mathbf{\Omega}, T\}$. Similarly, $\mathbf{x}_{B_i,j}(t)$ denotes agent j's state in agent i's body reference frame. Furthermore, $\boldsymbol{\lambda}$ and $\dot{\boldsymbol{\lambda}}$ describe the line-of-sight angle and angular rate vector, respectively. δt indicates the command cycle interval and $\tau_{perc,i}$ and $\tau_{inert,i}$ indicate the perception and inertial sensors (e.g. camera/IMU) tracking time delay respectively, reported for both agents in Table I. Tilde indicates that these inputs are subject to sensor bias and variance. Importantly, notice that sensor bias and variance is introduced at state level (i.e. $\tilde{\mathbf{x}}$), Equation 2) instead of introducing it explicitly later in the observation vector of an agent (i.e. $\tilde{\mathbf{o}}_i(t) = \mathbf{o}_i(t) + \mathcal{N}(\mu_{\mathbf{o}}, \Sigma_{\mathbf{o}})$). Hence, this (sensor) noise propagates through the computation (function $\mathrm{O}(\cdot)$) of these variables, causing a cross-correlated structure between entries in the observation vector, $\tilde{\mathbf{o}}_i(t)$. The limitation is that no auto-correlation structure is present in the noise.

In the previous definition, the observation function $\mathrm{O}(\cdot)$ selects the specific entries from the system state at the relevant time steps. In addition, it contains the computation for the LoS angles and the rate vector defined as,

$$\begin{aligned}\boldsymbol{\lambda}_i &= [\lambda_{yx,i}, \lambda_{xz,i}, \lambda_{yz,i}] \\ &= [atan2(y_{j-i}, -x_{j-i}), \\ &\quad atan2(-x_{j-i}, z_{j-i}), \\ &\quad atan2(y_{j-i}, z_{j-i})]\end{aligned} \tag{5}$$

and

$$\dot{\boldsymbol{\lambda}}_i = \frac{\mathbf{d}_{j-i} \times \mathbf{v}_{j-i}}{||\mathbf{p}_{j-i}||^2} = [\dot{\lambda}_{x,i}, \dot{\lambda}_{y,i}, \dot{\lambda}_{z,i}]. \tag{6}$$

where $\mathbf{d}_{j-i}$ and $\mathbf{v}_{j-i}$ define the position and velocity difference vector between agent i and j respectively. Notice that the angle vector is computed with respect to the body z-axis, as movement towards the target will require aligning thrust by tilting of this axis. This computation is inspired by the axes convention for missiles, with the x-axis oriented through the front of the missile and aligned with the source of thrust[4].

Motion Camouflage game mechanism
In nature, sophisticated hunters such as dragonflies are able to reduce the visual cues perceptible by their targets; thereby attempting to achieve *motion camouflage* (MC)[3]. In motion camouflage, the imminent approach of the pursuer is practically imperceivable to the evader, as it achieves a mimicry

of the background's optical flow and consequently appears stationary with respect to the evader's perceptive focal point [31]. Besides reducing visual cues, a pursuit trajectory achieving motion camouflage is desired because it can be proven to minimize interception time, zero-effort-miss distance and energy expenditure[4][32][33]. Formally, this game state can be described as,

$$\Gamma(t) = \left(\frac{\mathbf{p}_{e-p}(t)}{||\mathbf{p}_{e-p}(t)||} \cdot \frac{\dot{\mathbf{p}}_{e-p}(t)}{||\dot{\mathbf{p}}_{e-p}(t)||} \right) = \frac{\frac{\mathrm{d}}{\mathrm{d}t}||\mathbf{p}_{e-p}(t)||}{||\frac{\mathrm{d}\mathbf{p}_{e-p}(t)}{\mathrm{d}t}||} \tag{7}$$

with Γ also referred to as the *range vector correlation* and is bounded between $[-1, 1]$. The values of 1 and -1 denote the two game states with formal/perfect motion camouflage, the former with an escaping evader and the latter with an approaching pursuer. These game states exhibit the aforementioned advantageous properties, because the only relative movement between agents is along the irrotational line-of-sight vector.

In this research, we attempt to determine the effect of this game state on the learned behavior of our agents by emphasizing the lack of perceivable visual cues. Instead of explicitly encouraging motion camouflage through the reward definition, we test this by introducing an implicit game mechanism, dubbed the motion camouflage (MC) mechanism. This mechanism is active whenever Γ drops below a threshold of -0.9 (i.e. $\Gamma_{thres} = -0.9$), interpretable as less-than-perfect motion camouflage by the pursuer. The slack with regard to a perfect score for the pursuer (i.e. $\Gamma = -1$) follows from an assumed imperfect perception system of the evader, exploited by hunting dragonflies achieving effective motion camouflage in practice [34]. Furthermore, note that this asymmetric mechanism only affects the evader's perception, because we assume that the pursuer can overcome this perception deficiency through prediction and sensor fusion.

Whenever the MC game mechanism is active, the pursuer subjects the evader to outdated (previous) information on its position (through $\boldsymbol{\lambda}$) and nullifies any information on its relative motion (through $\dot{\boldsymbol{\lambda}}$).As long as the mechanism's condition is not violated, the evader's information on the pursuer's state is not updated, seemingly appearing relatively stationary over consecutive time steps. Implementation is achieved through a recursive definition of observation delay due to motion camouflage ($\tau_{MC_{t,e}}$) applied at the command frequency (δt),

$$\tau_{MC_e}(t) = \begin{cases} \tau_{MC_e}(t - \delta t) + \delta t & \text{if } \Gamma(t) \leq \Gamma_{thres} \\ 0 \quad (= reset) & \text{otherwise} \end{cases} \tag{8}$$

which is used to compute the observation set of the evader according to Equation 4 now defined as,

$$\tilde{\mathbf{o}}_e(t) = \mathrm{O}\left(\tilde{\mathbf{x}}_{B_e,e}(t), \tilde{\mathbf{x}}_{B_e,p}\big(t - \tau_{MC_e}(t)\big), \mathbf{u}_e(t); \delta t, \tau_{perc,e}, \tau_{inert,e} \right) \odot \mathbf{b}_{MC_t}(t), \tag{9}$$

where $\mathbf{b}_{MC}(t)$ represents an element-wise binary masking vector used to set the $\dot{\boldsymbol{\lambda}}$ entries to zero if the mechanism is active, implying no relative motion of the pursuer. Finally, note that the inclusion of this mechanism is expected to obstruct the policy estimation process for the evader, as it disrupts the relationship between inputs and outputs through the insertion of false observation samples. Effectively, this means that for the evader an additional uncertainty aspect is introduced to the already partially-observed game state.

D. Optimization

The aim of our optimization is to allow our policies to formulate robust strategies through exploration of *the* most influential game states; those that can rapidly impact the game's outcome. To this end, we formulate three design principles of our optimization routine. Firstly, we intend to decrease overall experiment wall-up time and increase sample throughput for our system in order to speed up optimization and to allow for the rapid evolution of policies. Secondly, encourage exploration combined with representative strategy feedback to limit prematurely stalled evolution encountered in a local optimum (i.e. the *red-queen* effect[35]). Finally, the promotion of *localized* sampling at these aforementioned most influential game states, primarily through the structure of the reward function.

Algorithm

The policies of both agents are individually optimized using the clipped Proximal Policy Optimization (PPO-clip) algorithm[36], implemented with default hyperparameters (unless specified in this report) using the *Ray RLlib*[37] Python library. The on-policy PPO algorithm is selected in an attempt to achieve high overall data throughput, cut down on the amount and size of our agents' neural networks as well as the associated memory required and reduce overall experiment wall-time compared to off-policy alternatives[38][39][40] such as MADDPG[41]. In addition, due to our fast simulator, sampling is considered inexpensive and a *replay buffer* might straggle in case of rapid evolution. In turn, this implies that with regard to the *multi-agent bias-variance tradeoff*[42], we opt for a reduction in bias and deliberately sacrifice (some) training stability. To promote continuous exploration and policy adaptability, we raise PPO's entropy coefficient to 0.001 and remove the Kullback–Leibler divergence loss. The on-policy nature ensures that newly explored strategies receive representative feedback, obtained by testing against adversaries that similarly optimize exclusively against the latest stage of the learning opponent.

Decentralized & Independent Learning

Our optimization setup is decentralized, implying two parallel systems optimizing their objectives independently. We do not opt for a centralized setup as we expect distinct *Value* scores and associated behavior at certain game states. We accredit this to the agents' asymmetric characteristics and conflicting objectives, further emphasized by their difference in reward structure defined at the end of this section. Furthermore, this choice allows us to monitor information on the optimization process of both agents separately (e.g. losses, entropy), deemed critical for the proper identification of issues at any stage of the optimization.

For this decentralized setup, we acknowledge that the partial and uncertain observation vector introduces additional task complexity that could hinder efficient convergence. Specifically, this relates to the estimation of the Value function, due to the potential non-uniqueness of $\{\boldsymbol{\lambda}, \dot{\boldsymbol{\lambda}}\}$ in the observation set at completely different game states. Hence, we append the observation vector for both agents' critics with entries containing additional information on the game state. On the other hand, the actor's observation set is left unchanged. For agent i the critic's complete observation set, $\tilde{\mathbf{o}}_{V,i}$, at time t is defined as,

$$\begin{aligned}\tilde{\mathbf{o}}_{V,i}(t) = \{&(1-\frac{t}{T}), d(t), ||\mathbf{v}_{j-i}(t)||, 1_{\tau_{MC_e}\neq 0}(t),\\ &\Gamma(t), \Gamma_{uV}(t), \Gamma_{PP_i}(t), \Gamma_{PP_j}(t)\}\end{aligned} \tag{10}$$

where $\tilde{\mathbf{o}}_i(t)$ indicates the previously defined noisy observation vector in Equation 4 originally or in case of an active motion camouflage mechanism in Equation 9. In order, the additional variables indicate the time left until truncation, distance, velocity difference norm, a *motion-camouflaged* pursuer indicator (subsection II-C) for the evader and (variants of) the range vector correlation (Γ, Equation 7). Note that the $1_{\tau_{MC_e}\neq 0}(t)$ variable is omitted for configurations with an inactive motion camouflage mechanism. Importantly, notice that all these additional introduced entries can be computed in the inertial reference frame and are provided to the critic with no time delay (even in case of no motion-camouflage mechanism). Please note that these inputs to the critic reflect our choice for offline reinforcement learning followed by online deployment of the actor. Most additional input elements to the critic described in Equation 10 would not be available in an online learning setting.

The latter Γ entries denote the unit-velocity (Γ_{uV}) and pure-pursuit (Γ_{PP}) range vector correlation coefficients defined as,

$$\Gamma_{uV}(t) = \frac{\mathbf{p}_{e-p}}{||\mathbf{p}_{e-p}||} \cdot \frac{\bar{\mathbf{v}}_e(t) - \bar{\mathbf{v}}_p(t)}{||\bar{\mathbf{v}}_e(t) - \bar{\mathbf{v}}_p(t)||} \tag{11}$$

and

$$\Gamma_{PP_i}(t) = \frac{\mathbf{p}_{j-i}(t)}{||\mathbf{p}_{j-i}(t)||} \cdot \frac{\mathbf{v}_i(t)}{||\mathbf{v}_i(t)||} \tag{12}$$

where $\bar{\mathbf{v}}_i$ denotes the normalized velocity vector of agent i. Compared to Γ, Γ_{uV} similarly describes relative motion of both agents with respect to the range vector, yet is invariant to velocity differences between agents. In contrast, it does not explicitly describe the rotation of the range vector and the minimum or maximum scores (1 & -1) do not describe motion camouflage states. Γ_{PP_i} describes to what degree agent i is aligning its own (not relative) motion with the range vector, with a maximum score of 1 conventionally known as the (greedy) *pure-pursuit* (PP) strategy. These versions of the range vector correlation coefficient are included to provide explicit information of the current individual and relative state of motion of both agents with respect to the range vector.

Rewards

In multi-agent reinforcement learning, attributing specific reward to agents' individual actions is notoriously complex, commonly referred to as the *credit assignment problem*[9][8]. In games of pursuit and evasion with reactive agents, this problem is especially apparent because it is hard to explicitly quantify how intermediary steps led to ultimate interception or escape. Additionally, this task is further complicated in our case as the asymmetric agents affect the distance vector is an unequal manner. On the other hand, these asymmetric influences can rapidly alter the game's outcome due to few (last) actions taken close to each-other, implying that proper sampling and optimization at these states becomes paramount. Hence, straightforward rewards definitions such as the distance at every time step or (solely) sparse rewards for game outcome do not recognize the asymmetric influence of agents and do not recognize this *localized* sampling urgency. Consequently, their use can hinder optimization (convergence) through the promotion of counterproductive actions due to incorrect definitions of the agents' contribution and the general *over-sampling* at less influential game states.

In consideration of these reward attribution complexities, this research proposes an asymmetric rather than a zero-sum reward structure, conventionally observed in pursuit evasion games[7]. To start, we consider that our game takes place in a boundless arena, but acknowledge that the impact of the agents' actions varies relative to the distance between each other (e.g. close by or far off each other). Therefore, we define a rescaled version of the *distance from interception* ($d - d_{INT}$) as,

$$\begin{aligned}D(d) = \frac{1}{c}\Bigg(&\frac{-1}{(d-d_{INT})^2 + a} + ((d-d_{INT}) + b)^2\\ &- (b+1)^2 + 1/a\Bigg)\\ \underset{\sim}{\propto} &-\frac{1}{(d-d_{INT})^2} + (d-d_{INT})^2\end{aligned} \tag{13}$$

where $a(= 10^{-1.5})$ and $b(= 3)$ are used to control the general slope in the $(d < 1)$ & $(d > 1)$ regions respectively and $c(= 100)$ is used to rescale the function to $[0, \approx 1]$ in the distance range of $[0, 5]$. Notice that in this function, the scalar a governs both the lower bound for the equation and ensures it is strictly larger than zero. An illustration of this monotonic scaling function is provided in Figure 2, where one can observe that the slope is most pronounced at the lower ($d < 0.5$) and higher ($d > 1.5$) end distances, purposefully designed to improve learning ability. Specifically, at $d \approx 0$ the $\frac{1}{d^2}$ term dominates with a steep decrease close to interception which offers an informative gradient here as well as the ability to offset behavior at distances further away. On the other hand, the d^2 dominates at larger distances to compensate for loss of gradients in $\frac{1}{d^2}$ and distinguish between games at $d \approx 1$ and $d >> 1$. For both terms a squared formulation, instead of a linear one, is chosen to further amplify differences in game state across the entire spectrum of distances.

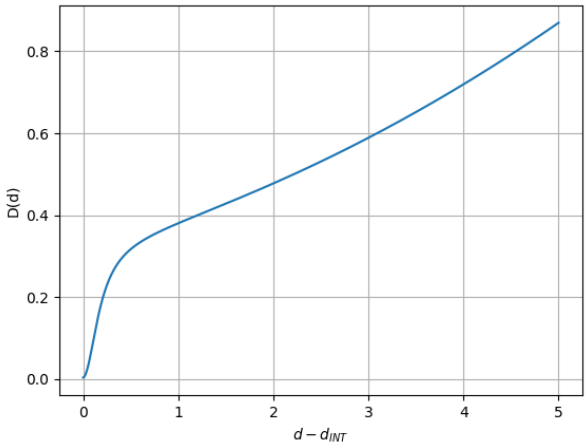


Fig. 2. Scaled version of distance to interception ($d - d_{INT}$) in [0,5] range for the monotonic function $D(d)$ in Equation II-D for $a = 10^{-1.5}, b = 3, c = 100$.

This scaled version of the distance is used to weigh actions at various game states on relative importance and is used to define agent specific rewards. Both agents receive equal but opposite rewards for an interception event and time taken. The rewards for the pursuer ($r_{t,p}$) and evader ($r_{t,e}$) agents are represented as,

$$\begin{aligned} r_{t,p} &= -\frac{10}{T}D(d_t) && -\frac{5}{T} + 5 \cdot 1_{INT} \\ r_{t,e} &= +\frac{10}{T}\frac{\big(D(d_t) - D(d^e_{t-1})\big)}{||\mathbf{p}_{t,e} - \mathbf{p}_{t-1,e}||} && +\frac{5}{T} - 5 \cdot 1_{INT} \end{aligned} \tag{14}$$

where 1_{INT} and T represent an interception indicator and the time step limit available for trials. $D(\cdot)$ denotes the distance scaling function (bounded to $[0, \approx 1]$) of and d_t the conventional distance metric between agents, defined by $||\mathbf{p}_{t,p} - \mathbf{p}_{t,e}||$. d^e_{t-1} represents the distance between the current pursuer and the previous evader position, defined by $||\mathbf{p}_{t,p} - \mathbf{p}_{t-1,e}||$.

For the pursuer, we select a unity discount factor (i.e. $\gamma = 1$) such that the agent considers the rewards for the entire remainder of the trial and how it contributes to its outcome. In combination with the relative weighting, the reward structure is designed to encourage both time reduction and an interception event without continuous greedy improvement in distance, yet with enough information to distinguish performance across trials with equal outcomes through the mean of the scaled distance ($\frac{D(d_t)}{T}$). The steep nature of the $\frac{1}{d^2}$ term in $D(d_t)$ (Equation II-D) now proves important, as it encourages consecutive interception attempts with increasingly better scores for nearer misses. Furthermore, it is important to recognize that this reward structure implies that the distance metric is the complete responsibility of the pursuer, which constitutes a formally incorrect reward allocation due to the evader's influence. However, recall from the agent characteristics in Table I that the pursuer can operate at a much higher speed, yet with lower maneuverability. Hence, its actions are generally expected to impact the distance the most, especially after overshooting of the target (i.e. miss) and subsequently due to its larger turning radius during recovery, visualized in Figure 3. Consequently, we deem the pursuer, attempting interception, to be in general command of the distance metric, implying that any slack thereof is interpretable as disadvantageous.

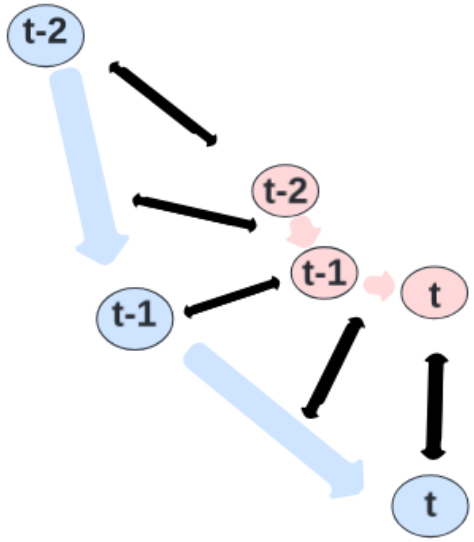


Fig. 3. Visualization of velocity differences between pursuer (blue) and evader (red) agents as reason for assigning pursuer the *distance responsible* agent and allowing evader to focus on its own distance change.

For the evader agent, we similarly focus on trial time and interception, but in an opposite manner. On the other hand, we focus on individual gains in the scaled distance measure through the $\frac{\big(D(d_t) - D(d^e_{t-1})\big)}{||\mathbf{p}_{t,e} - \mathbf{p}_{t-1,e}||}$ term, computed by considering the current position of the pursuer and the relative position of the evader at the current and previous time step. The difference between these two time steps in the numerator (i.e. $D(d_t) - D(d^e_{t-1})$), serves as an imperfect proxy for the individual contribution of the evader to the distance metric and is visualized in Figure 4. Importantly, recognize that this value is not equal to and should not be interpreted as the difference in distance between two consecutive time steps (i.e. $(D(d_t) - D(d_{t-1}))$), which is influenced by both agents. In addition, the use of the scaled distance rather than a simpler distance measure means that rewards for gains in distance are conditional on the game state and most pronounced close to the interception threshold, where maneuvering steps are deemed most influential on the game's outcome.

Consequently, a beneficial property arises from both agents' reward descriptions with regard to *localized* sampling efficiency, as a higher frequency of these encounters is encouraged through the pursuer's reward function; theoretically promoting action sampling at these influential states near the interception distance. Moreover, the division by the total displacement of the evader in between consecutive time steps (i.e. $\frac{1}{||\mathbf{p}_{t,e} - \mathbf{p}_{t-1,e}||}$) is used to avoid minimization of optimization loss by a reduction of speed.

Finally, the evader is designed to be more short-sighted/greedy than the pursuer through a γ discount factor of 0.99, which was selected to speed up reward improvement and resemble the stressed rather than purely-strategic behavior expected from chased insect targets.

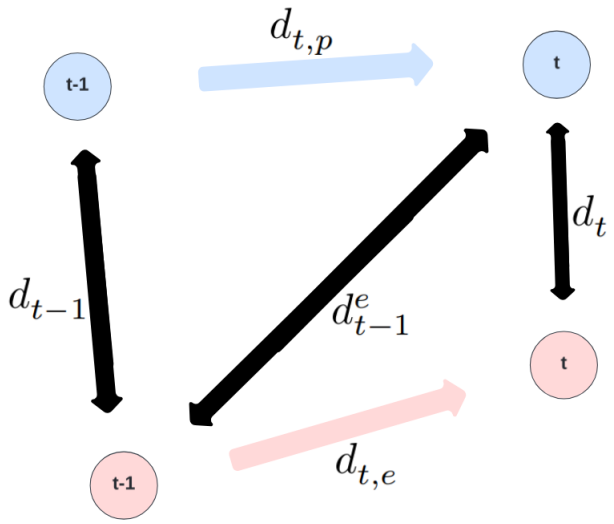


Fig. 4. Illustrative example for turn-based approximation of evader's individual contribution to distance change (i.e. d^e_{t-1}) by comparison between current and previous position of the evader (red) with respect to the current pursuer (blue, top-right) position. In addition, it contains the definition of the evader displacement $d_{t,e} = ||\mathbf{p}_{t,e} - \mathbf{p}_{t-1,e}||$.

Hall-of-Fame

Due to the adaption flexibility of our policies, we recognize that our optimization setup might be prone to cyclic evolution in which agents repetitively formulate non-innovative strategies, rather than presenting absolute improvement[43]. Alternatively, the setup might be too flexible for agents to steadily formulate comprehensive strategies at all, consequently failing to produce sustained growth.

Therefore, in this research we wish to assess the adaptive flexibility of our setup on the optimization of policies and the formulation of agent strategies. We test this by comparing the results from optimization with and without a *Hall-of-Fame* (HoF) sampling mechanism. With an active HoF mechanism, sampling of an episode/trial occurs with an on-policy pursuer confronting either an on- or off-policy evader. The off-policy evaders comprise the minority of adversaries for the pursuer and originate from historical policies (i.e. earlier optimization iterations). It is important to note that optimization of policies remains exclusively based on the on-policy samples of the respective agent. This always holds for the pursuer, yet is critical to emphasize for the evader; since its sampled set now contains samples from both types.

Although the mechanism only adjusts the evader's sampling, both agents are expected to be affected. If active, the on-policy pursuer optimizes against a broader range of adversaries and is consequently expected to attribute relatively less attention to the latest on-policy evader. Related to the aforementioned *multi-agent bias/variance tradeoff* [42], this theoretically introduces a bias or *evolutionary lag* in the favor of a potential gain in optimization stability and robustness to cyclic evolution. For this same reason, the evader might face simpler and/or faster optimization, as the evader might gain an *evolutionary lead* due to the potentially constrained evolutionary flexibility of the pursuer. These potential effects clearly resemble characteristics of multi-agent off-policy algorithms such as MADDPG[41], yet is expected to maintain higher policy adaptability due to the exclusive optimization with regard to on-policy samples. Hence, an active mechanism can be interpreted as an attempt at implementing off-policy algorithm characteristics into our on-policy setup, while retaining the remainder of our optimization design choices.

The Hall-of-Fame is implemented at the sampling stage of every optimization iteration, with the current iteration denoted as K. At this stage, we select H cached historical evader policies selected at equidistant iterations in the range [0, K). The *focus probability*, p_{HoF}, governs the frequency of a confrontation against the current on-policy evader or the set of off-policy evaders across sampling trials. The historical policies are chosen uniformly at a frequency of $\frac{1-p_{HoF}}{H}$. Similar to the on-policy versions, the historical policies are stochastic instead of deterministic, because the mechanism affects the generation of samples meant for optimization. In our case, we select policies at equidistant iterations between $[0, K]$, opt for five historical policies ($H = 5$) and set our focus probability at 75% ($p_{HoF} = 0.75$). As an example, for $K = 10$ this implies historical policies at $k = \{0, 2, 4, 6, 8\}$ sampled each at 5% of the trials, summing up to the remaining 25%.

E. Policies

Actors & Critics

The model configurations for our actor and critic networks describe commonly observed network architectures applied in time series analysis and control tasks[44]. For the actor network, we utilize a recurrent Long Short-Term Memory (LSTM) neural network[45] with 64 units and access to the 10 last observation sets (i.e. $\{\tilde{\mathbf{o}}_{i,t-9}, \cdots, \tilde{\mathbf{o}}_{i,t}\}$). After the LSTM layer, the transformed final set of features passes through a linear layer to obtain the eight outputs, comprising the four means and four logarithmic standard deviations required to parameterize the Gaussian distribution ($\mathbf{u}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{I}_4\boldsymbol{\sigma}_i)$) representing agent i's stochastic action policy $\pi_i(a_{i,t}|o_{i,t})$. For the critic network, we utilize a LSTM model with 32 units connected to two subsequent layers of 16 units with ReLU activation and a final layer with a single output representing the value estimate, $\hat{V}(\mathbf{o}_{V,i,t})$. Similar to the actor, the value network's LSTM has access to the last 10 time steps and subsequent layers are applied to the final time step of features. With a command frequency set at 100 Hz, they both have access to an observation set spanning the most recent 0.1 seconds (i.e. $0.01 \cdot 10$), albeit that they have different observation delay characteristics as reported in Table I.

Behavioral cloning

In our research, we implement behavioral cloning for the pursuer's actor model to obtain the agent's initial policy through imitation learning of another pre-trained pursuer, denoted as the *expert* policy, capable of intercepting non-reactive (i.e. dummy) moving targets. The reasons for this are threefold. First of all, behavioral cloning of all actor models under consideration from the same pre-trained actor aligns their initial capabilities and invokes consistency for subsequent analysis. Secondly, the initial training phase is expected/observed to devote substantial training time to the *learning to fly* problem[29] with the objective of maintaining stability and preventing crashing, which does not form the main focus of this research. In contrast, behavioral cloning is not applied to the evader model because there is no potential of crashing due to the absence of gravity. In addition, the learning of initial evasive maneuvers is not deemed critical for the evader, since the pre-trained pursuer is only optimized for a lower interception rate. Finally, recall from Equation 14 that the agent's reward definitions are conditional on the game state through the scaling distance function ($D(d)$, Equation II-D). This function reflects that game states close to the interception distance are most influential. Therefore, it is important that action sampling takes place at these states to allow for a proper and focussed optimization process, as stated in the introduction of subsection II-D. Behavioral cloning encourages this *localized* sampling efficiency, as the initial actor starts with pursuit trajectories in the vicinity of the evader.

Behavioral cloning through imitation learning is implemented by action mismatch minimization between the expert (exp) and the pursuer's policies through supervised learning, with the error defined as $e_t = (a_{p,t} - a_{exp,t})^2$ [46]. This approach forces the initial policy's mean to align with that of the expert formulated as,

$$\pi_{p,k=0}(a_{p,t}|o_{p,t}) \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p \mathbf{I}_4) \rightarrow \mathcal{N}(\boldsymbol{\mu}_{exp}, \boldsymbol{\sigma}_p \mathbf{I}_4) \quad (15)$$

where both $\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p$ are outputs of the policy's parameterized model. k indicates the optimization iteration, where 0 defines the initial parameter set. Notice that while the distribution's mean parameter is adjusted, we force the policy's standard deviation to remain unchanged. The latter is conducted to prevent premature exploitation in the subsequent main optimization phase with both agents.

We implement the methodology by Ferede et al. [22] and obtain the expert behavioral policy by optimizing a pursuer model with our LSTM neural network architecture with our observation vector, our pursuer agent characteristics and our game environment initialization. In addition, we adjust the task focus from gate-passing to the interception of a non-reactive (dummy) target moving along a straight line with constant velocity magnitude uniformly sampled from $||\mathbf{v}|| \sim \mathcal{U}(0.5; 3)$. For this task, we employ the pursuers reward function defined in Equation 14 and optimize for this single-agent setting using vanilla PPO with default hyperparameters and a discount factor of 0.99. The command frequency is set at 100 Hz and trials end due to truncation at 1 second (i.e. 100 steps) or termination due to an intercept at 15 cm (i.e. $d_{INT} = 0.15$) and train until we achieve an interception rate of $\approx 25\%$. This interception distance is deliberately three times higher than during the main optimization phase ($d_{INT} = 0.05$) and the optimization is intentionally ended at this lower interception rate. This is done in order to obtain a pursuer model that is clearly capable of approach, but not precisely honed. Once again, this is in consideration of the aforementioned ambition to improve *localized* sampling efficiency at the most critical game states near interception during the main optimization phase.

III. Experimental setup

In this research, we analyze simulation experiments in order to determine the hunting strategy and efficacy of our pursuer's policy, which is considered the principal agent of this research. In our analysis, we compare pursuer policies originating from our multi-agent deep reinforcement learning optimization with a set of benchmarks.

Optimization of our deep reinforcement learning setup runs for a maximum of 300 iterations, with a single experiment taking approximately 14 hours (wall-time) using 8 CPU workers for parallel environment sampling and 1 NVIDIA GeForce GTX1660Ti GPU for optimization. Every optimization iteration comprises a sampling stage, an optimization stage and an evaluation stage. During the sampling stage of every iteration, 100 trials with a maximum duration of 10 seconds are simulated for our pursuit-evasion scenario according to the agents' stochastic policies. During the optimization stage of every iteration, the model is fitted exclusively on the data sampled stochastically at this iteration (i.e. no data leakage from previous iterations) and is evaluated in a deterministic setting after fitting. During the evaluation stage of every iteration, 100 trials with a maximum duration of 10 seconds are simulated for our pursuit-evasion scenario according to the agents' deterministic commands by always opting for the mean of the action distribution implied by the agents' policies.

Over these iterations, the policies do not remain consistent, as they continuously adapt with regard to the behavior of the adversary. Nonetheless, general periods of convergence can be observed across configurations by tracking game metrics, where we focus on the highest pursuer interception rate and test for (temporary) stationarity through the Kwiatkowski–Phillips–Schmidt–Shin (KPSS)[47] test, with null hypothesis for a stationary time series. Upon no rejection of the KPSS null hypothesis at a 5% significance level, we evaluate our policies over this range and compare results to policies originating from other configurations and/or benchmarks. These comparisons are statistically assessed by utilizing the Mann-Whitney U test[48] for continuous metrics and McNemar's test[49] in case trials' outcomes (i.e. interception or escape) can be directly compared between configurations as in the to-be-defined *offline* setting.

Evaluation and analysis is conducted for two pursuit-evasion scenarios, dubbed the *online* and *offline* setting. In the online setting, the pursuer's policy is tested against the adversarial evader's policy, both optimized through the methods described in section II. In contrast to this, the offline setting describes

the confrontation between the optimized pursuit controller and recordings of undisturbed flights of the Opogona moth, the primary target species of this study, provided by PATS. PATS is a Delft-based agro-tech startup that develops autonomous drone systems for pest monitoring and control in greenhouses[50]. A representative visualization is provided in Figure 1. Importantly, the optimization process for the pursuit controller has not considered the Opogona recordings before this evaluation stage. To clarify, note that in both simulation settings, we invoke deterministic outcomes from the policy's stochastic distribution by always opting for its mean controller output in order to attain consistency in our evaluation.

Both settings are investigated in order to provide a more complete overview of the pursuit controller's capabilities in the absence of real-life testing, by addressing different limitations of our methodology/analysis. The online setting provides insight into the controller's pursuit capabilities and interception efficacy against a reactive evader. On the other hand, the offline setting illustrates these same capabilities with respect to a non-reactive agent, but one with actual insect dynamics. Hence, it illustrates controller generalizability to previously unseen evader system dynamics and addresses the limitation that the evader model is only a qualitative match to insect's capabilities, raised in subsection II-B.

Furthermore, for the online setting, we analyze the agents' aggregate behavior over optimization iterations and across configurations through comparison of the statistics based on mean game metrics collected across trials within an iteration. Thus, the analysis considers differences in configurations across sets of iterations. In contrast, the offline setting compares configurations through a more detailed approach with analysis of behavior across trials. This is achieved for the offline setting by selecting a single representative policy for the pursuer agent at a specific optimization iteration, dubbed the *focus* iteration, to confront the non-reactive Opogona recording and by subsequent tracking of progression of the game metrics over the remainder of the trial. Moreover, the same procedure is applied to compare behavior of agents originating from different configurations directly, through simulated confrontations of these policies. This latter setting is considered *online* as well.

Benchmarks

In this research, we compare our investigated configuration to a triplet of pursuer benchmarks. First of all, we define the *drones versus drones* or *DRONES* benchmark, which is used to determine the effect of using an insect-inspired model. This benchmark comprises a confrontation between the quadcopter pursuer and a similar quadcopter evader, instead of an insect-inspired one. We implement this by utilizing the same characteristics for the evader as for the pursuer defined in Table I, yet with $g_e = 0$ and $T_e = [0, g]$. The former adjustment is made in line with the discussion in subsection II-B; in order to bound the game to a stable z-coordinate. The latter comprises a reduced thrusting ability of the evader in order to account for the associated absence of gravity.

Secondly, we define the *Behavioral Cloning Expert* or *BCE* benchmark, which serves as a starting point to compare the effect of subsequent design choices (e.g. multi-agent optimization, evader reward, insect-inspired dynamics, etc). Notably, it extends the two-dimensional study by Rano [51] to a three-dimensional case, who investigates the use of single-agent reinforcement learning for non-reactive target interception. Specifically, this benchmark implements the methodology described at the end of subsection II-D, with a non-reactive evader moving with constant velocity along a straight line. In contrast to the approach there, we do not terminate optimization prematurely, set interception distance at 5 cm and run trials for the full 10 seconds.

Third and finally, we define the *Custom Proportional Navigation* or *CPN* benchmark, which is used to determine the effect of utilizing our nonlinear parametric controllers which operate on the target and drone state as well as in consideration of the vehicle dynamics. This CPN benchmark reflects a custom implementation of the classical proportional navigation guidance law. Importantly, this control law is not specifically designed for the vehicle at hand and provides controls unconditional on the current state of the pursuer drone. This benchmark is implemented in the absence of gravity for the quadcopter model described in Equation 3. It comprises a proportional-derivative controller on the line-of-sight angle and rates and is defined as,

$$\begin{aligned} \Omega_{cmd,p}(t) &= -0.3\tilde{\boldsymbol{\lambda}}(t-\tau_{perc_p}) + 3\tilde{\dot{\boldsymbol{\lambda}}}(t-\tau_{perc_p}) \\ T_{cmd,p}(t) &= ||\tilde{\mathbf{v}}_{e-p}(t-\tau_{perc_p})|| \\ &\quad \cdot ||\tilde{\mathbf{p}}_{e-p}(t-\tau_{perc_p}) \times \Omega_{cmd,p}(t)|| \\ T_{cmd,p}(t) &\approx ||\mathbf{a}_{cmd,p,CPN}|| \end{aligned} \tag{16}$$

where $\boldsymbol{\lambda}$ and $\dot{\boldsymbol{\lambda}}$ are defined in Equation 5 and Equation 6 respectively. Tilde denotes that the signals are subject to noise according to the methodology associated with Equation 4. τ_{perc_p} denotes the perception delay set in Table I. $\bar{\mathbf{p}}_{e-p} \times \Omega_{cmd,p}(t)$ denotes the cross product between normalized range vector and angular velocity commands. $||\mathbf{v}_{e-p}||$ denotes the velocity difference norm.

This benchmark comprises a custom implementation of the classical proportional navigation law for angular rate-thrust control instead of the more conventional acceleration commands (i.e. $\mathbf{a}_{cmd,CPN}$). The reason for this is to maintain command structure consistency to the investigated configurations. Therefore, the thrust setting comprises the norm of the acceleration commands of the conventional guidance law[4]. The gain on $\dot{\boldsymbol{\lambda}}$ has been set at 3 based on manual tuning following empirical results and comprises a common setting in missile guidance [4] as well as insect predators[28]. In contrast to conventional PN, the $\boldsymbol{\lambda}$ component is introduced to improve the stability of the guidance law and has been tuned manually as well. Finally, note that the proportional navigation guidance law and this custom implementation are both theoretically known to and empirically observed to diverge after misses[4][31][52]. Therefore, in our implementation we re-initialize the attitude of our controller to align with the normalized range vector ($\tilde{\bar{\mathbf{p}}}_{e-p}$) whenever the range to the target exceeds 3 m, according to the initialization methodology

described at the end of subsection II-B. This measure is interpretable as a turning maneuver implemented by an additional controller and is implemented to allow for fair analysis to our nonlinear parameterized controllers, which are capable of learning recovery after misses.

IV. Results & Discussion

In this research, we investigate the ability of a multi-agent deep reinforcement learning setup applied to a pursuit and evasion scenario to identify effective interceptors and to what extent this setup replicates natural behavior. In this section, we perform comparative analysis on three levels; focussing on identifying the general viability of the proposed approach as well as determining the effect of game/agent mechanisms and reward definitions.

The research is evaluated in the *online* and *offline* settings as introduced in section III. For both these settings, we perform an ablation study to investigate the effect of our *motion-camouflage* game mechanism (*MC*, subsection II-C) and the *Hall-of-Fame* sampling mechanism (*HoF*, subsection II-D) on our optimization process and our agents' strategies. Furthermore, we compare our base configuration with our custom-designed dense reward structure (Equation 14) to a sparser zero-sum game definition under a unity discount factor with only the defined rewards components for time-taken and interception, dubbed the *ZEROSUM* configuration. Moreover, we consider performance to our benchmark configurations, *CPN*, *BCE* and *DRONES* as described in section III. Finally, note that while the analysis in this chapter is mostly performed at an aggregate level, the agents' specific trajectories for every trial can be viewed at https://github.com/rwvosTUD/DragonfliesAndDrones.git for both our evaluation settings and across our investigated configurations.

For our three configurations and the DRONES benchmark, the interception rate as a function of the optimization iteration is visualized in Figure 5. Specifically, the interception rate is defined as for the confrontations between agents from the same configuration, evaluated in the *online* setting. As described in section III, for our analysis we focus on specific sub-ranges within the optimization iterations that exhibit general stability with regard to tracked game metrics. We identify these stable sub-ranges by iteratively searching for the largest range that still upholds the KPSS null hypothesis. For these ranges, we provide and analyze the mean and standard deviation statistics of various game and agent specific metrics and evaluate their difference through testing in subsequent sections. For more detailed analysis at trial level for both the online and offline settings, we choose a *focus* iteration of the pursuer's policy within this range, to represent the general strategy and capabilities of the agent at this stage/range. This is done by selecting the iteration closest to the median interception rate observed in this interval. For our three configurations and the DRONES benchmark, these stable regions are described in Table II alongside associated interception rate statistics. In addition, the start and end of these stable regions are visualized in Figure 5 through the dashed vertical axes.

A. Game statistics

For our investigated configurations, the statistics of the trial average metrics across iterations are reported in Table III, alongside a visualization of the interception rate in Figure 5. For this figure, we can identify differences in trends and convergence rate through the exponentially smoothed moving average traces, yet conclude that these differences are nominal in consideration of the consistently large variance of the individual traces. On the other hand, note that while initial divergence across configurations is observable in Figure 5, the traces ultimately converge towards a final, stable stage at the end of the optimization routine.

Following these observations, we consider Table III which comprises various aggregate statistics from the selected stable ranges of Table II that can quantify fundamental differences in game scenarios between configurations. In general, the results from Table III imply similarity of the simulated games between configurations; the differences in metrics are statistically significant according to the Mann-Whitney U-tests compared to the base configuration, yet the numeric differences in mean and standard deviation do not emphasize substantial qualitative dissimilarity. Specifically, they suggest that games between agents are comparable in distance between agents ($\bar{d}$), the trial time to total time available ($\bar{T}$), the velocity ratio between agents ($\overline{\mathbf{v}_p/\mathbf{v}_e}$). Moreover, they do not highlight clear and explicit variation in strategies performed by the agents on this aggregate level because of the comparable scores on the range vector correlation ($\bar{\Gamma}$) and ratio of time the evader is blind due to a motion-camouflage pursuer ($\bar{1}_{MC,e}$). The exception to these remarks is the DRONES configurations, which presents a significantly lower velocity ratio; implying nearly similar speeds. For this configuration, these agents share nearly symmetric characteristics in terms of agility and top speed, which is observed to result in tail-chases at high speeds during trials rather than more sophisticated evasive maneuvers (visualized at the GitHub repository) in an attempt to improve the evader's survival time.

Except for the DRONES configuration, these aforementioned observations imply that our optimization setup is not prone to experience vastly distinct types of games, but games where the agents are expected to ultimately formulate similar behavior/strategies. An adjusted configuration specification (e.g. an active MC mechanism) might bring forth different behavior, but that type of behavior still generally resembles that of the base configuration. Importantly, this is not to imply that the learned agents' policies are directly interchangeable, yet that differences in performance are expected to be attributed to improved implementation of similar strategies rather than the implementation of completely alternative ones.

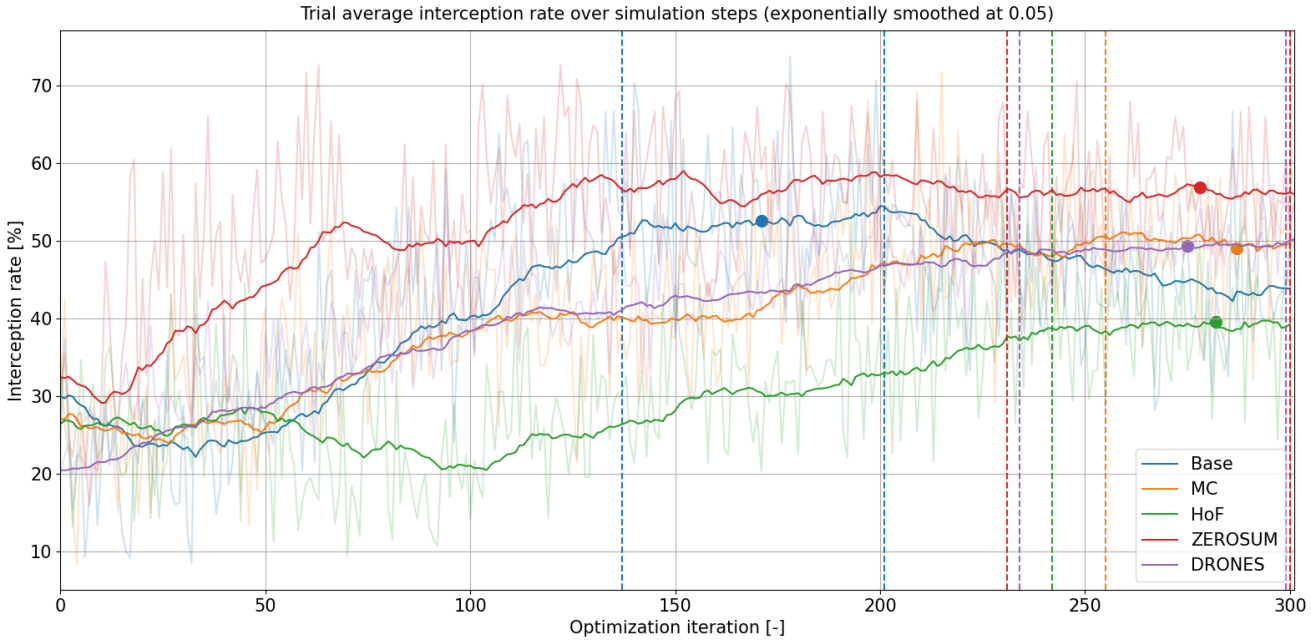


Fig. 5. Mean interception rates over optimization iterations across configurations, exponentially smoothed by a factor of 0.05. The dashed vertical axes highlight the start and end of the presumed 'stationary'/stable periods used to compute the statistics reported in Table III. Within this range, the KPSS test is performed and a focus iteration is selected, closest to the median interception rate which is denoted by a large circular dot in its respective color. Range descriptions and associated interception statistics correspond to those reported in Table II.

Name	$k_{[\text{start,end}]}$	$1_{INT,(\text{min,med,max})}$	p_{KPSS}	k_{focus}	$1_{INT,\text{focus}}$	$1^*_{INT,\text{online}}$	$1^*_{INT,\text{offline}}$
Base	[137, 201]	(0.377, 0.532, 0.738)	0.10	171	0.53	0.60	0.92
MC	[255, 299]	(0.321, 0.500, 0.587)	0.11	287	0.49	0.52	0.93
HoF	[242, 300]	(0.244, 0.396, 0.559)	0.13	282	0.40	0.44	0.94
ZEROSUM	[231, 300]	(0.429, 0.564, 0.706)	0.11	278	0.57	0.60	0.85
DRONES	[234, 299]	(0.382, 0.491, 0.603)	0.12	275	0.50	0.51	0.39

TABLE II

START AND END POINT OF OUR CONVERGED ITERATION (k) RANGE ALONGSIDE THE MINIMUM, MEDIAN AND MAXIMUM INTERCEPTION RATE (1_{INT}) AS WELL AS THE KPSS P-VALUE (p_{KPSS}) FOR THE INTERCEPTION RATE WITHIN THE SELECTED RANGE. RANGES ARE VISUALIZED IN FIGURE 5. WITHIN THIS RANGE, A FOCUS ITERATION IS SELECTED CLOSEST TO THE MEDIAN RATE WHICH IS USED FOR DETAILED EVALUATION OF BOTH THE ONLINE AND OFFLINE SETTINGS, WITH RESPECTIVE INTERCEPTION RATES PROVIDED IN THE LAST TWO COLUMNS.* DENOTES THE INTERCEPTION RATES ASSOCIATED WITH THE VISUALIZATIONS AVAILABLE AT THE GITHUB REPOSITORY. NOTE THAT $1^{INT,online}$ AND $1^*_{INT,online}$ ORIGINATE FROM SEPARATE CONFRONTATION EXPERIMENTS WITH A DISCREPANCY IN RATES DUE TO RANDOMNESS IN TRIAL INITIALIZATION.

Combining these insights, the difference in level between configurations in Figure 5 at the aforementioned final, seemingly stable, stage of optimization is to be attributed to the asymmetric impact of the introduced mechanisms on the game. For the MC mechanism, the interception rate is expected to increase because it introduces an uncertain aspect to the already partially observable game state for the evader. For the HoF mechanism, the pursuer is at the disadvantage, because it needs to focus on multiple versions of the evader. Compared to the ZEROSUM configuration, the potential effect is expected to be less explicit and harder to quantify because the reason for adjusting the reward function was to improve the agents' general learning capability rather than the identified strategy. Nonetheless, without the additional dense terms in the reward function (Equation 14) the ZEROSUM pursuer might be at the advantage as the ZEROSUM evader might struggle to effectively learn a strategy due to the aforementioned credit assignment problem (subsection II-D).

B. Online Confrontations

After comparing aggregate game statistics in their respective stable ranges, we consider the trial average game statistics from confrontations between configurations at their respective *focus* iteration (see Table II) provided in Table IV. Examining this table, we generally observe no significant differences between the performance and strategy statistics of the base, MC and HoF pursuers confronting the base configuration's evader, with an interception rate up to 55%. Correspondingly, the reverse cases (i.e. base pursuer versus alternative evaders) do not suggest otherwise.

Pursuer	**Evader**	1_{INT}	$\bar{d}$	$\bar{1}_{d<0.15}$	$\bar{T}$	$\overline{\mathbf{v}_p/\mathbf{v}_e}$	$\bar{\Gamma}$	$\bar{1}_{MC,e}$
Base	Base	0.53 (0.08, -)	0.69 (0.04, -)	0.07 (0.01, -)	0.68 (0.06, -)	5.07 (0.25, -)	-0.31 (0.05, -)	0.15 (0.05, -)
MC	MC	0.5 (0.07, 0.05)	0.81 (0.05, 0.0)	0.06 (0.01, 0.0)	0.7 (0.05, 0.05)	5.85 (0.27, 0.0)	-0.29 (0.05, 0.02)	0.17 (0.05, 0.01)
HoF	HoF	0.39 (0.07, 0.0)	0.76 (0.03, 0.0)	0.05 (0.0, 0.0)	0.77 (0.05, 0.0)	4.27 (0.3, 0.0)	-0.24 (0.03, 0.0)	0.14 (0.05, 0.42)
ZEROSUM	ZEROSUM	0.56 (0.07, 0.02)	0.65 (0.05, 0.0)	0.08 (0.01, 0.0)	0.66 (0.05, 0.05)	3.52 (0.19, 0.0)	-0.28 (0.04, 0.0)	0.13 (0.04, 0.01)
DRONES	DRONES	0.5 (0.06, 0.01)	0.74 (0.08, 0.0)	0.1 (0.01, 0.0)	0.64 (0.05, 0.0)	1.36 (0.07, 0.0)	-0.27 (0.06, 0.0)	0.12 (0.05, 0.0)

TABLE III

SUMMARIZING STATISTICS REPRESENTING THE MEAN AND STANDARD DEVIATION OF TRIAL AVERAGE STATISTICS OBSERVED ACROSS ITERATIONS FOR OUR INVESTIGATED CONFIGURATIONS, WITHIN THE STABLE ITERATION RANGE AS PROVIDED IN TABLE II. EVERY ITERATION WITHIN THIS RANGE CONSIDERS 100 TRIALS WHICH ARE AGGREGATED INTO AN ITERATION MEAN WHICH IN TURN IS AGGREGATED ACROSS ITERATIONS TO PRODUCE THE STATISTICS PRESENTED IN THIS TABLE. TO ADDRESS THE DIFFERENCE BETWEEN CONFIGURATIONS, THE SECOND TERM IN BRACKETS DENOTES THE P-VALUE OF THE MANN-WHITNEY U TEST FOR ALL COLUMNS, OBTAINED BY COMPARING THE DENOTED CONFIGURATION'S METRICS TO THOSE OF THE BASE CONFIGURATION. STARTING AT THE RIGHT OF THE INTERCEPTION RATE, THE STATISTICS IN ORDER DENOTE THE TRIAL AVERAGE FOR THE METRICS; DISTANCE, DURATION IN THE NEAR-MISS RANGE ($d < 0.15$), TOTAL DURATION, VELOCITY RATIO, RANGE VECTOR CORRELATION, AND DURATION THAT THE PURSUER IS MOTION-CAMOUFLAGED RESPECTIVELY.

Pursuer	**Evader**	1_{INT}	$\bar{d}$	$\bar{1}_{d<0.15}$	$\bar{T}$	$\overline{\mathbf{v}_p/\mathbf{v}_e}$	$\bar{\Gamma}$	$\bar{1}_{MC,e}$
Base	Base	0.55 (-)	0.7 (0.27, -)	0.08 (0.06, -)	0.64 (0.38, -)	5.19 (1.69, -)	-0.35 (0.32, -)	0.12 (0.33, -)
MC	Base	0.51 (-)	0.8 (0.31, 0.02)	0.06 (0.03, 0.0)	0.7 (0.36, 0.54)	5.6 (2.63, 0.24)	-0.29 (0.31, 0.17)	0.14 (0.35, 0.68)
HoF	Base	0.54 (-)	0.74 (0.33, 0.56)	0.07 (0.04, 0.1)	0.69 (0.35, 0.6)	5.74 (2.19, 0.11)	-0.29 (0.28, 0.58)	0.18 (0.39, 0.24)
ZEROSUM	Base	0.44 (-)	0.75 (0.3, 0.23)	0.06 (0.04, 0.0)	0.72 (0.36, 0.22)	5.82 (2.18, 0.03)	-0.28 (0.29, 0.21)	0.14 (0.35, 0.68)
DRONES	Base	0.18 (-)	1.48 (0.77, 0.0)	0.03 (0.04, 0.0)	0.86 (0.32, 0.0)	8.06 (2.71, 0.0)	-0.1 (0.32, 0.0)	0.06 (0.24, 0.14)
BCE	Base	0.21 (-)	1.29 (0.81, 0.0)	0.03 (0.03, 0.0)	0.83 (0.34, 0.0)	7.72 (3.0, 0.0)	-0.17 (0.32, 0.0)	0.07 (0.26, 0.23)
Base	MC	0.55 (-)	0.72 (0.31, 0.9)	0.06 (0.03, 0.02)	0.67 (0.35, 0.61)	5.48 (2.33, 0.87)	-0.3 (0.27, 0.51)	0.21 (0.41, 0.09)
Base	HoF	0.50 (-)	0.73 (0.25, 0.63)	0.06 (0.06, 0.0)	0.7 (0.32, 0.56)	5.64 (2.18, 0.58)	-0.28 (0.24, 0.34)	0.15 (0.36, 0.54)
Base	ZEROSUM	0.56 (-)	0.59 (0.2, 0.0)	0.1 (0.07, 0.05)	0.65 (0.37, 0.98)	3.23 (1.11, 0.0)	-0.29 (0.28, 0.17)	0.2 (0.4, 0.12)
Base	DRONES	0.03 (-)	1.97 (0.54, 0.0)	0.01 (0.02, 0.0)	0.97 (0.15, 0.0)	0.96 (0.19, 0.0)	0.39 (0.27, 0.0)	0.02 (0.14, 0.01)
Base	BCE	0.99 (-)	0.54 (0.12, 0.0)	0.12 (0.06, 0.0)	0.23 (0.18, 0.0)	2.38 (0.61, 0.0)	-0.69 (0.31, 0.0)	0.17 (0.38, 0.32)

TABLE IV

SUMMARIZING STATISTICS REPRESENTING THE MEAN AND STANDARD DEVIATION OF TRIAL AVERAGE STATISTICS OBSERVED ACROSS THE EVALUATED 100 CONFRONTATIONAL TRIALS BETWEEN INVESTIGATED CONFIGURATIONS AT THE SELECTED *focus* POLICY ITERATION (TABLE II). TO ADDRESS THE DIFFERENCE BETWEEN CONFIGURATIONS, THE SECOND TERM IN BRACKETS DENOTES THE P-VALUE OF THE MANN-WHITNEY U TEST FOR ALL OTHER COLUMNS THAN THE FIRST, OBTAINED BY COMPARING THE DENOTED CONFIGURATION'S METRICS TO THOSE OF THE BASE CONFIGURATION. NOTE THAT UNLIKE FOR THE OFFLINE RESULTS IN TABLE V, THE MCNEMAR TEST CANNOT BE PERFORMED FOR THE FIRST COLUMN IN THIS CASE AS THE TRIALS CANNOT BE COMPARED DIRECTLY. STARTING AT THE RIGHT OF THE INTERCEPTION RATE, THE STATISTICS IN ORDER DENOTE THE TRIAL AVERAGE FOR THE METRICS; DISTANCE, DURATION IN THE NEAR-MISS RANGE ($d < 0.15$), TOTAL DURATION, VELOCITY RATIO, RANGE VECTOR CORRELATION AND DURATION THAT THE PURSUER IS MOTION-CAMOUFLAGED RESPECTIVELY.

On the other hand, we observe a substantial reduction of 20% for the interception rate of the ZEROSUM's pursuer versus the base evader, yet consistent performance for the reverse case. This result underscores the intention of the dense reward structure for the agents of the base configuration in Equation 14 and indeed improves the base evader's escape capabilities. Specifically, we observe that games with a ZEROSUM evader play out at closer distances (note $\bar{d}$ in Table IV and also Table III), suggesting that these evaders under-develop the urgency of keeping the pursuer at bay. For the base configuration, we focus on this aspect explicitly in its reward definition of the evader and, in turn, also see that the pursuer optimizes against this class of evader to develop a strategy robust to both cases.

Besides the investigated configurations, we also compare the base evader's performance against benchmark pursuers. We observe the DRONES and BCE pursuers confronting the base evader end up at roughly half/one-third of the base pursuers interception rate. Both cases serve as evidence against the generalizability of these pursuit policies in the capture of reactive evaders with insect-inspired dynamics. Where the BCE pursuer optimizes interception of a non-reactive evader moving along a straight trajectory, the drones' pursuer optimizes interception of a reactive evader with similar drone-like dynamics. Hence, these results underscore that the base configuration's evader characteristics are distinctly different and that the selection of the evader has substantial impact on the identified pursuer policies resulting from our optimization routine. Finally, notice that through comparison of these two benchmarks the effect of a reactive evader (versus non-reactive alternative) is not immediately evident; we jointly address this aspect with the results of offline evaluation setting in the subsequent section.

Furthermore, the reverse cases show that the base pursuer is capable of intercepting a non-reactive evader moving along a straight trajectory (i.e. the BCE evader), but clearly fails at intercepting an evader with nearly symmetric (drone-like) characteristics (i.e. the DRONES evader). Compared to DRONES pursuer versus Base evader, the latter observation implies that the DRONES pursuer generalizes better to an insect-like evader than the base pursuer does to a DRONES evader. This conclusion is mainly attributed to the speed at which the agents have learned to operate, visualized in the confrontations available at the GitHub repository. While for the DRONES configuration the pursuer learns to operate at high(er) speed to ultimately catch the evader in a tail-chase, for the base configuration the asymmetry in agent characteristics has led the evader to learn to maneuver rather than force a detrimental tail chase. Hence, the base pursuer is not able

Pursuer	**Evader**	1_{INT}	$\bar{d}$	$\bar{1}_{d<0.15}$	$\bar{T}$	$\overline{\mathbf{v}_p/\mathbf{v}_e}$	$\bar{\Gamma}$	$\bar{1}_{MC,e}$
Base	offline	0.92 (-)	0.69 (0.25, -)	0.08 (0.04, -)	0.27 (0.31, -)	3.43 (4.03, -)	-0.49 (0.35, -)	0.16 (0.37, -)
MC	offline	0.93 (-, 1.0)	0.74 (0.25, 0.23)	0.07 (0.04, 0.02)	0.28 (0.3, 0.55)	3.49 (3.8, 0.68)	-0.46 (0.34, 0.8)	0.15 (0.36, 0.85)
HoF	offline	0.94 (-, 0.79)	0.71 (0.19, 0.48)	0.08 (0.03, 0.19)	0.29 (0.29, 0.4)	3.66 (3.64, 0.15)	-0.43 (0.34, 0.3)	0.15 (0.36, 0.85)
ZEROSUM	offline	0.85 (-, 0.12)	0.71 (0.29, 0.75)	0.07 (0.03, 0.18)	0.37 (0.35, 0.05)	3.44 (2.96, 0.36)	-0.4 (0.33, 0.16)	0.13 (0.34, 0.55)
DRONES	offline	0.39 (-, 0.0)	1.59 (0.97, 0.0)	0.04 (0.04, 0.0)	0.66 (0.45, 0.0)	5.14 (2.68, 0.0)	-0.27 (0.4, 0.0)	0.06 (0.24, 0.02)
BCE	offline	0.42 (-, 0.0)	1.36 (1.18, 0.0)	0.04 (0.04, 0.0)	0.7 (0.41, 0.0)	4.65 (4.0, 0.0)	-0.18 (0.32, 0.0)	0.12 (0.33, 0.42)
CPN	offline	0.19 (-, 0.0)	1.82 (0.53, 0.0)	0.02 (0.03, 0.0)	0.85 (0.33, 0.0)	13.41 (41.88, 0.0)	-0.28 (0.26, 0.0)	0.11 (0.31, 0.3)

TABLE V

SUMMARIZING STATISTICS REPRESENTING THE MEAN AND STANDARD DEVIATION OF TRIAL AVERAGE STATISTICS OBSERVED ACROSS THE EVALUATED 100 OFFLINE TRIALS FOR OUR INVESTIGATED CONFIGURATIONS AT THE SELECTED *focus* POLICY ITERATION (TABLE II). TO ADDRESS THE DIFFERENCE BETWEEN CONFIGURATIONS, THE SECOND TERM IN BRACKETS DENOTES THE P-VALUE OF THE MCNEMAR TEST FOR THE FIRST COLUMN AND THE MANN-WHITNEY U TEST FOR ALL OTHER COLUMNS, OBTAINED BY COMPARING THE DENOTED CONFIGURATION'S METRICS TO THOSE OF THE BASE CONFIGURATION. STARTING AT THE RIGHT OF THE INTERCEPTION RATE, THE STATISTICS IN ORDER DENOTE THE TRIAL AVERAGE FOR METRICS; DISTANCE, DURATION IN THE NEAR-MISS RANGE ($d < 0.15$), TOTAL DURATION, VELOCITY RATIO, RANGE VECTOR CORRELATION AND DURATION THAT THE PURSUER IS MOTION-CAMOUFLAGED RESPECTIVELY.

to catch up to the DRONES evader, attempting to move away at straighter trajectories with increasing speed. On the other hand, the faster DRONES pursuer strikes at the evader with occasional success similar to the base configuration, yet also loses a lot more time in recovery compared to the base configuration in case of a miss.

C. Offline Opogona recordings

We evaluate our pursuers' strategies at the selected *focus* iterations (Table II) in the offline setting and provide the aggregate statistics across trials in Table V. While the evaluation in the offline setting comprises non-reactive targets, it does provide insight in the pursuer's ability to generalize to realistic dynamics of the intended insect target.

Examining Table V, we observe the highest interception rate of the Opogona recordings for our base, MC and HoF configurations, with an interception rate of up to 94%. Amongst these configurations, we note no significant differences in the game metrics similar to the results in Table IV. On the other hand, we observe a reduction in interception rate for the ZEROSUM configuration, yet cannot establish its significance according to the McNemar test. These opposing statements imply that the alignment is still stronger than the misalignment for this evaluation attempt. Inline with subsection IV-B, we would attribute the improved interception rate to the improved ability of the evader to optimize its strategy through the more informative dense reward component in Equation 14, which in turn requires the pursuer to improve as a consequence of the dynamic and conflicting nature of our game scenario.

While comparison to the ZEROSUM case addresses the proposed reward function's design, comparison of our base configuration to our benchmarks allows us to further conclude on other aspects of our design. Compared to the benchmarks (CPN, BCE & DRONES - section III) with an interception rate of up to 42%, we observe substantial and significant improvement of the interception rate for our base configuration in Table V. Through comparison of the base configuration to our three benchmarks controllers, we draw up three main conclusions about our simulation setup based on this table. First of all, comparison of the base configuration to both the DRONES and BCE benchmarks suggest that it is required to consider the target's characteristics. In addition, it verifies the intended qualitative match between the design of our insect-inspired evader in the base, MC & HoF configurations and the true Opogona for this non-reactive setting to a certain degree.

Secondly, the additive effect of optimization against a reactive evader is not clearly evident from the results. Similar to the results in Table IV, we observe comparable performance in aggregate game metrics between the DRONES and BCE benchmarks. Comparison of these benchmarks to the base configuration implies an evader model with insect-inspired dynamics is an impactful design choice, but not that this evader needs to be reactive. Thus, while a reactive evader is expected in practice, the results considered in this report do not clearly establish the need for multi-agent optimization. Since, the multi-agent optimization setup in our research introduces numerous design complexities, its implementation should not be done arbitrarily. Hence, further analysis into this perspective is recommended through practical testing in order to address the need for this design choice, which falls outside the scope of the current research.

Third and finally, the parameterized nonlinear controllers investigated in this research highlight their ability to improve interception efficacy through consideration of the specific quadcopter vehicle dynamics at hand. Compared to all other configurations and benchmarks, the CPN controller attains the lowest interception rate in Table V, at roughly half the rate of more parameterized benchmarks (BCE & DRONES). Recall from section III that this controller has access to the same target observation (through λ and $\dot{\lambda}$) as the parameterized controllers, yet forms a fixed and linear combination to set its reference. While previous research has identified performance deterioration in the face of observation noise and delay for proportional navigation-based guidance[53][54][55], they do not predict considerable failure of strategy implementation at our investigated levels of noise and delay (Table I).

Instead, we attribute the improved performance of our parameterized pursuers to their optimized control of the specific INDI quadcopter vehicle model defined in Equation 3. This aspect is already apparent by comparing the CPN and the BCE benchmarks. While the BCE benchmark was only trained with a simplified evader, it does have the ability to dynamically set

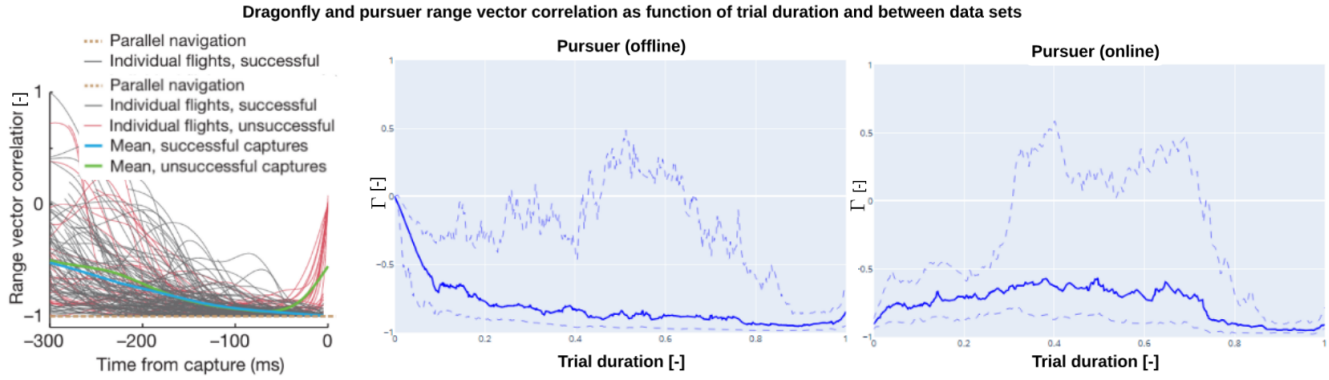


Fig. 6. Range vector correlation (Γ, Equation 7) for trajectories of real dragonfly hunting prey (left, 112 successful flights, image retrieved from [34]), the base configuration's pursuer hunting the Opogona recordings (center, 92 successful flights, top row Table V) and the base configuration's pursuer hunting the base evader (right, 55 successful flights, top row Table IV). For the center and right image, blue lines indicate the 75^{th} (dashed top), 50^{th} (bold) and 25^{th} (dashed bottom) percentiles of Γ observed across flights respectively.

a specific control reference (i.e. $\{\mathbf{\Omega}_{cmd}, T_{cmd}\}$) conditional on the quadcopter's state (i.e. $\{\mathbf{\Omega}, T\}$) and its vehicle dynamics. On the contrary, the CPN benchmark comprises a fixed control law, which sets references unconditional on the quadcopter's state. Subsequently, by comparing BCE with the base configuration in Table IV and Table V it is clear that further optimizing this commanded reference becomes even more important under increased task complexity (e.g. insect-inspired and/or reactive evader) as maximization of the quadcopter's capabilities is required to improve strategy implementation and maintain interception efficacy. Alternatively, one might tackle the incorporation of commands (currently conducted by INDI) rather than improving commands itself to maximize the quadcopter's capabilities. To this end, one might extend this study by considering the potential of end-to-end controllers such as those by Ferede et al.[22] to circumvent limitations that any higher-frequency inner-loop controller such as INDI might exhibit.

Lastly, notice that the benchmark configurations operate at significantly higher velocity ratios than the multi-agent configurations, with the CPN benchmark reaching the maximum. Once again, we attribute this to higher pursuer speed associated with the fact that these benchmarks have not been optimized for an asymmetric opponent in the form of a maneuverable evader with insect like dynamics. The multi-agent configurations can be observed (in visualizations at the GitHub repository) to reduce their speeds in an attempt to retain more agility themselves and be less sensitive to sudden evader trajectory adjustments as well as achieving faster recovery after a missed attempt. The benchmark configurations have not been tuned with regard to these asymmetric agent characteristics and attempt interception after building up speed during straight(er) trajectories and consequently also experience slower recovery. The CPN benchmark does worst in this aspect; showing the least flexibility in adapting to evader maneuvers and with no recovery ability other than the hard reset (defined in section III). Hence, it builds up considerable speed during the trials.

D. Trial progression

In the remainder of this chapter we analyze our strategies as the trial progresses in order to achieve insight into the real-time behavior of our agent during the interception attempt. Two sets of 100 online and offline trials have been collected at the *focus* iterations described in Table II. Since we cannot analyze the 3D trajectories in this report directly, we focus on the distance (d) and (variants of) the range vector correlation coefficient (Γ) to assess both the general game state as well as the agents' motion with regard to the line-of-sight vector at various stages of the trial, respectively.

Figure 6 and Figure 7 visualize the average conventional range vector correlation coefficient (Γ, Equation 7) and average pure-pursuit range vector correlation coefficient (Γ_{PP}, Equation 12) over the duration, respectively. In Figure 7 we notice that our pursuer persistently closely aligns its own motion with the line-of-sight vector. Subsequently, in Figure 6 we further observe that this strategy consistently achieves a negative Γ score close to -1 in both evaluation settings, signifying that the relative velocity also aligns with said vector. This score indicates that our pursuer exploits its superiority in speed to continuously implement an approach trajectory that counteracts the evader's motion and establishes an irrotational line-of-sight vector[52]. Comparing the online and offline scores, we observe that the online case faces greater deviation from the $\Gamma = -1$ boundary. This is attributed to the online evader actively attempting to escape its pursuer , observable by the steady negative score for the pure-pursuit range vector correlation of the evader (i.e. $\Gamma_{PP,e}$) in Figure 7, indicating its attempt at increasing the range vector. In addition, notice that this metric is close to -0.6 rather than -1 which means that the evader opts for an escapee trajectory that does not strictly align with the line-of-sight vector. This implies that the evader implements maneuvering motion rather than forcing a tail chase (implied by score closer to -1), which can be expected for the evader due to its inferior top speed (subsection II-B).

Through comparison with the left plot in Figure 6, we

Fig. 7. Pure-pursuit range vector correlation (Γ_{PP}, Equation 12) for successful online (top, 55 successful trials, top row Table IV) and offline (bottom, 92 successful trials, top row Table V) trajectories of the base configuration for the pursuer (left column) and evader (left column) respectively. For the images, blue lines indicate the 75^{th} (dashed top), 50^{th} (bold) and 25^{th} (dashed bottom) percentiles of Γ_{PP} coefficients observed across flights respectively.

observe the Γ scores for our pursuer align with natural behavior exhibited by hunting dragonfly[34][1]. This characteristic is desired in a pursuit strategy, as it theoretically achieves time-optimality, energy minimization and motion camouflage[31][33][32]. Importantly, recall that this behavior was not explicitly encouraged through the reward definition as described in subsection II-D (i.e. through minimization of Γ). Hence, this result highlights the ability of our reinforcement learning setup to mimic natural behavior for our pursuer subject to imperfect and minimalistic information. The latter statement, related to the observation set in Equation 4, is interesting by itself as performant behavior is not guaranteed for a controller subject to noise and delay as described by Raju et al.[55].

On the other hand, the resemblance between the pursuit controllers investigated in this report and dragonflies is limited. In nature, dragonflies' interception efficacy can be attributed to their physical traits (agility and speed)[1], careful prey selection based on heuristics[2], hunting from below and against a cluttered background from the prey's perspective and the implementation of motion camouflage during pursuit[34]. The latter conditions encourage a predictable prey trajectory, which forms an important aspect in the choice to pursue[56] and means dragonfly can hunt along sophisticated *proactive* trajectories resembling proportional navigation. In comparison to the setting in this research, the exploitation of true observational blind-spots of the adversary is not possible and information can only be limited through achieving and maintaining motion camouflage. This is a difficult feat to maintain, so our evader is generally aware of the pursuer and acts to limit its success accordingly. As a result, our controllers become *reactive* rather than *proactive*, diverging from behavior exhibited by dragonflies exploiting their prey's information deficiencies. This is observed through the (near) maximum scores on $\Gamma_{PP,p}$ in Figure 7 implying consistent adherence to pure pursuit. In this divergence, our controllers more closely resemble fly species such as blowflies, who also hunt reactive and erratic targets without said information deficiencies [57][58].

V. Limitations & Future Research

This research has provided insight into the interception efficacy of a drone pursuer with regard to a more maneuverable evader and now considers open questions for future research. While the observation set comprises a biologically plausible one, the unlimited field-of-view of both our agents is unrealistic. In fact, as discussed in subsection IV-D, for natural

hunters such as dragonfly, consistent tracking of the target on the visual field forms an important factor in the consideration to continue/start pursuit[2]. Similarly, this plays a crucial role in the ultimate interception attempt, where in nature both types of agents can be observed to attempt to exploit the blind spot of their adversaries[1][6]. Consequently, it is expected that although limiting the field-of-view will further complicate optimization attempts through reduced observability, its implementation might further replicate natural behavior and provide insight into the feasibility of the interception strategies in autonomous pursuer drones with onboard visual perception.

Furthermore, the current report does not shed light on the pursuit controller's feasibility under more challenging circumstances. For the pursuer, this study is limited to simplified implementations of factors such as observation noise, motor dynamics and the overall implementation capability of the controller-proposed strategy. Consequently, it is recommended to further extend the study by probing the controller's limitations and identifying failure conditions, similar to the theoretical study by Raju et al.[55]. For the evader, it is evident that the vehicle model is only a qualitative rather than quantitative match, where an improved replication would allow for an optimized pursuer to be better prepared for its true adversary. Moreover, one might obtain an improved replication of the evader through an exercise in system identification of the Opogona recordings and by specifically considering escape behavior closest to interception, as described by Corcoran et al.[6] in the evader's reward definition.

Moreover, recall that our analysis in section IV does not provide decisive answers on the effectiveness of utilizing a reactive evader (i.e. multi- vs single-agent optimization) as well as the potential of improved pursuer vehicle models. The expected reactive nature of evaders in practice has led to a multi-agent optimization routine for this research, yet we deem practical testing required to accurately address the perks of our specific design implementation thereof as well as its need in general. In addition, our analysis also showed that classical linear benchmarks emphasize the ability of our nonlinear parameterized controllers to set references conditional on the target and drone state as well as the drone vehicle model. Hence, one might expand the study in an end-to-end fashion similar to Ferede et al.[22] to circumvent limitations that any higher-frequency inner-loop controller such as INDI might encounter with regard to strategy implementation.

Finally, from a game theoretical perspective it is currently unclear as to what extent the identified controllers comprise a local stable optima or a global equilibrium state such as a *Nash equilibrium*, which describes the state at which the strategies are optimized against the worst possible behavior of the adversary. Analysis along this theoretical avenue might further highlight shortcomings of the pursuit controller and allow for subsequent improved designs.

VI. Conclusion

This research has sought out to contribute to the design of autonomous artificial hunters capable of consistent interception of insect pests. Therefore, we have optimized the adversarial strategies of two agents in a differential game of pursuit and evasion through multi-agent deep reinforcement learning. In addition, we propose parameterized controllers for dedicated vehicle models with asymmetric capabilities in speed and maneuverability, further subject to minimalistic sets of biologically-plausible observations. As a whole, our methodology addresses the complexities of optimizing multiple adversarial agents in this scenario with regard to informative sampling , individual credit attribution and evolutionary adaptability.

From our results, we show that our quadcopter pursuer is consistently able to pursue and intercept both a reactive and more maneuverable insect-inspired evader as well as recordings of actual insect targets, achieving interception rates of 55% and 94% on these respective tasks. In comparison, pursuers alternatively optimized against non-reactive evaders or reactive drone-like evaders with symmetric capabilities, achieve an interception rate of only 42% for the same insect target recordings. Despite these promising results, we conclude that further research is needed to formally establish the superiority of multi-agent optimization in this asymmetric game scenario. Furthermore, we observe that our pursuer mainly implements pure-pursuit as well as motion camouflage to some degree; drawing comparison to the hunting strategy of dragonfly. However, further emphasizing this behavior through an implicit game mechanism does not impact the controller's learned behavior. Moreover, optimizing the pursuer through confrontations of both the current evader's policy and previous ones through an explicit Hall-of-Fame mechanism also does not impact the optimization outcome significantly. In addition, we observe that our proposed insect-inspired evader model provides a qualitative match to actual insect dynamics and that it allows our pursuer to formulate a strategy for the consistent capture of offline recordings. On the other hand, we do not find decisive evidence for the need of a reactive evader and conclude that further research is needed to formally establish the superiority of multi-agent optimization in this asymmetric game scenario. Through comparison of our parameterized controllers to classical alternatives, we attribute implementation success to their ability to set control references conditional on drone and target state, especially in consideration of the sluggish vehicle model at hand. Finally, following discussion on the limitations of the approach in a theoretical setting, we recommend the need for practical testing in order to provide a decisive answer to the feasibility of the proposed solution.

References

[1] R. J. Bomphrey, T. Nakata, P. Henningsson, and H.-T. Lin, "Flight of the dragonflies and damselflies," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 371, no. 1704, p. 20150389, 2016.

[2] H.-T. Lin and A. Leonardo, "Heuristic rules underlying dragonfly prey selection and interception," *Current Biology*, vol. 27, no. 8, pp. 1124–1137, 2017.

[3] A. Mizutani, J. S. Chahl, and M. V. Srinivasan, "Motion camouflage in dragonflies," *Nature*, vol. 423, no. 6940, pp. 604–604, 2003.

[4] N. A. Shneydor, *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998.

[5] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace sciences*, vol. 91, pp. 99–131, 2017.

[6] A. J. Corcoran and W. E. Conner, "How moths escape bats: predicting outcomes of predator–prey interactions," *Journal of Experimental Biology*, vol. 219, no. 17, pp. 2704–2715, 2016.

[7] I. E. Weintraub, M. Pachter, and E. Garcia, "An introduction to pursuit-evasion differential games," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1049–1066.

[8] S. V. Albrecht, F. Christianos, and L. Schäfer, "Multi-agent reinforcement learning: Foundations and modern approaches," *Massachusetts Institute of Technology: Cambridge, MA, USA*, 2023.

[9] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019.

[10] X. Qiu, P. Lai, C. Gao, and W. Jing, "Recorded recurrent deep reinforcement learning guidance laws for intercepting endoatmospheric maneuvering missiles," *Defence Technology*, vol. 31, pp. 457–470, 2024.

[11] W. Wang, M. Wu, Z. Chen, and X. Liu, "Integrated guidance-and-control design for three-dimensional interception based on deep-reinforcement learning," *Aerospace*, vol. 10, no. 2, p. 167, 2023.

[12] X. Fu, J. Zhu, Z. Wei, H. Wang, and S. Li, "A uav pursuit-evasion strategy based on ddpg and imitation learning," *International Journal of Aerospace Engineering*, vol. 2022, pp. 1–14, 2022.

[13] B. Vlahov, E. Squires, L. Strickland, and C. Pippin, "On developing a uav pursuit-evasion policy using reinforcement learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 859–864.

[14] B. Gaudet, R. Furfaro, and R. Linares, "Reinforcement learning for angle-only intercept guidance of maneuvering targets," *Aerospace Science and Technology*, vol. 99, p. 105746, 2020.

[15] X. Gong, W. Chen, and Z. Chen, "Intelligent game strategies in target-missile-defender engagement using curriculum-based deep reinforcement learning," *Aerospace*, vol. 10, no. 2, p. 133, 2023.

[16] C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4552–4559, 2021.

[17] K. Wan, D. Wu, Y. Zhai, B. Li, X. Gao, and Z. Hu, "An improved approach towards multi-agent pursuit–evasion game decision-making using deep reinforcement learning," *Entropy*, vol. 23, no. 11, p. 1433, 2021.

[18] J. Ye, Q. Wang, B. Ma, Y. Wu, and L. Xue, "A pursuit strategy for multi-agent pursuit-evasion game via multi-agent deep deterministic policy gradient algorithm," in *2022 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE, 2022, pp. 418–423.

[19] H. Xiong, H. Cao, and W. Lu, "A dynamics perspective of pursuit-evasion games of intelligent agents with the ability to learn," in *2022 41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 7082–7087.

[20] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[21] L. Sun, Y.-C. Chang, C. Lyu, Y. Shi, Y. Shi, and C.-T. Lin, "Toward multi-target self-organizing pursuit in a partially observable markov game," *Information Sciences*, vol. 648, p. 119475, 2023.

[22] R. Ferede, G. de Croon, C. De Wagter, and D. Izzo, "End-to-end neural network based optimal quadcopter control," *Robotics and Autonomous Systems*, vol. 172, p. 104588, 2024.

[23] E. J. Smeur, Q. Chu, and G. C. De Croon, "Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 450–461, 2016.

[24] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.

[25] S. Li, E. Öztürk, C. De Wagter, G. C. De Croon, and D. Izzo, "Aggressive online control of a quadrotor via deep network representations of optimality principles," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6282–6287.

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[27] P. T. Gonzalez-Bellido, S. T. Fabian, and K. Nordström, "Target detection in insects: optical, neural and behavioral optimizations," *Current opinion in neurobiology*, vol. 41, pp. 122–128, 2016.

[28] S. T. Fabian, M. E. Sumner, T. J. Wardill, S. Rossoni, and P. T. Gonzalez-Bellido, "Interception by two predatory fly species is explained by a proportional navigation feedback controller," *Journal of The Royal Society Interface*, vol. 15, no. 147, p. 20180466, 2018.

[29] J. Eschmann, D. Albani, and G. Loianno, "Learning to fly in seconds," *IEEE Robotics and Automation Letters*, 2024.

[30] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[31] E. W. Justh and P. Krishnaprasad, "Steering laws for motion camouflage," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, no. 2076, pp. 3629–3643, 2006.

[32] P. Glendinning, "The mathematics of motion camouflage," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 271, no. 1538, pp. 477–481, 2004.

[33] N. Carey and M. Srinivasan, "Energy-efficient motion camouflage in three dimensions," *arXiv preprint arXiv:0806.1785*, 2008.

[34] M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, and A. Leonardo, "Internal models direct dragonfly interception steering," *Nature*, vol. 517, no. 7534, pp. 333–338, 2015.

[35] L. Van Valen, "The red queen," *The American Naturalist*, vol. 111, no. 980, pp. 809–810, 1977.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[37] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 3053–3062.

[38] N. Heess, D. Tb, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[39] X. Lyu, Y. Xiao, B. Daley, and C. Amato, "Contrasting centralized and decentralized critics in multi-agent reinforcement learning," *arXiv preprint arXiv:2102.04402*, 2021.

[40] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[41] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[42] J. G. Kuba, M. Wen, L. Meng, H. Zhang, D. Mguni, J. Wang, Y. Yang *et al.*, "Settling the variance of multi-agent policy gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 458–13 470, 2021.

[43] S. Nolfi, "Co-evolving predator and prey robots," *Adaptive Behavior*, vol. 20, no. 1, pp. 10–15, 2012.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[47] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

[48] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[49] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[50] P. I. D. Solutions, "Monitor and eliminate pests in your greenhouse crops," *PATS Website*, 2024.

[51] I. Rañó, "On motion camouflage as proportional navigation," *Biological cybernetics*, vol. 116, no. 1, pp. 69–79, 2022.

[52] P. Reddy, E. W. Justh, and P. Krishnaprasad, "Motion camouflage in three dimensions," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 3327–3332.

[53] K. S. Galloway, E. W. Justh, and P. S. Krishnaprasad, "Motion camouflage in a stochastic setting," in *2007 46th IEEE conference on decision and control*. IEEE, 2007, pp. 1652–1659.

[54] P. Reddy, E. W. Justh, and P. S. Krishnaprasad, "Motion camouflage with sensorimotor delay," in *2007 46th IEEE conference on decision and control*. IEEE, 2007, pp. 1660–1665.

[55] V. Raju and P. Krishnaprasad, "Motion camouflage in the presence of sensory noise and delay," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2846–2852.

[56] T. J. Wardill, S. T. Fabian, A. C. Pettigrew, D. G. Stavenga, K. Nordström, and P. T. Gonzalez-Bellido, "A novel interception strategy in a miniature robber fly with extreme visual acuity," *Current Biology*, vol. 27, no. 6, pp. 854–859, 2017.

[57] N. Boeddeker, R. Kern, and M. Egelhaaf, "Chasing a dummy target: smooth pursuit and velocity control in male blowflies," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. 1513, pp. 393–399, 2003.

[58] M. Land, "Chasing and pursuit in the dolichopodid fly poecilobothrus nobilitatus," *Journal of Comparative Physiology A*, vol. 173, pp. 605–613, 1993.

Part III

Literature Review

1 Introduction **24**

2 Strategies for pursuit and interception **25**

2.1 Three strategies . . . 25
2.2 Motion camouflage . . . 29
2.3 Mixing MCPG to ensure interception . . . 36
2.4 Final remarks . . . 38

3 Games of pursuit and evasion **39**

3.1 Differential games . . . 39
3.2 Deep Reinforcement Learning methods . . . 40
3.3 Insights from implementations . . . 44
3.4 Final remarks . . . 45

4 Neural pursuit controllers **46**

4.1 Artificial neural networks . . . 46
4.2 Liquid gated synapses . . . 48
4.3 Liquid time-constant networks . . . 49
4.4 Neural Circuit Policies . . . 51
4.5 LTC-NCP implementations . . . 52
4.6 Final remarks . . . 54

5 Conclusion **54**

5.1 Challenges and Opportunities . . . 54
5.2 Conclusion . . . 55
5.3 Future Research . . . 55

6 Literature review tables **57**

*This part has been assessed for the course AE4020 Literature Study.

Biologically-inspired controller design and optimization for pursuit and interception

R.W. Vos, *r.w.vos@student.tudelft.nl*
Faculty of Aerospace Engineering, Delft University of Technology

Abstract - ***This literature review explores the design of autonomous controllers for micro-air vehicles (MAVs) inspired by natural predators like dragonflies. It evaluates interceptive pursuit strategies from a control perspective and analyses how their properties contribute to interception success. Following this evaluation, the use of deep reinforcement learning and recurrent neural networks for robust controller identification in differential games of pursuit and evasion scenarios is determined as a method to optimize nonlinear parameterized controllers. With regard to such controllers, this review highlights the potential of liquid networks with neural circuit policies as scalable alternatives to conventional RNNs for the task at hand. All in all, this review paves the way for future research with the aim to replicate natural predator behavior through the use of bio-inspired neural pursuit controllers optimized through simulated games of pursuit and evasion.***

Index terms - Differential games of pursuit and evasion, Simulated Evolution, Multi-agent Deep Reinforcement Learning, Proportional Navigation, Motion Camouflage, Recurrent neural networks, Liquid time-constant networks, Neural Circuit Policies.

1 Introduction

Micro-air vehicles (MAVs) with autonomous control have become ever more present in society. In agriculture, these systems are already being used to help improve yields through detailed monitoring of crops. Although these methods can help to improve awareness in real-time, they do not offer direct solutions to combat harmful entities such as insect pests, conventionally countered through the use of insecticide. A sustainable alternative to reduce this need for insecticides and offer farmers more control would be to eliminate these pests through interception by a MAV. However, designing autonomous controllers capable of competitive pursuit and consistent interception in consideration of potential evader reactions is no small feat. Hence, one might consider the continuously evolving behavior of expert hunters in nature as inspiration for this type of autonomous control.

Natural predators such as dragonfly implement sophisticated pursuit strategies with consistently high interception rates, reaching upwards of 80%[2]. By studying how these insects and other predators execute precise and efficient hunts, researchers can extract principles that inspire the design of artificial controllers. This perspective gives rise to a cross-disciplinary approach for the development of autonomous systems which merges biology and engineering, leading to the identification of control laws and game theoretic tactics for the use in pursuit and evasion scenarios such as missile guidance. In line with these results, this literature review intends to summarize and structure key findings from these disciplines in order to formulate effective design principles for autonomous controllers capable of the task at hand. Consequently, the main research question of this literature review is,

How can insights from nature, game theory and robotics be used to design controllers capable of implementing robust interceptive strategies for the aerial pursuit of reactionary evaders?

In order to achieve robust interception of similarly evolved reactionary evaders, autonomous control onboard MAVs requires advanced control systems capable of real-time decision-making and maneuvering. Throughout the last decade, the potential of deep reinforcement learning algorithms and novel artificial intelligence models has been established. Therefore, the previous research question also considers how one might leverage these technologies to design the autonomous controllers.

In order to address this research topic effectively, the main question is further divided into three subquestions which are introduced in separate chapters which address the following three aspects. Firstly, in Chapter 2 it is considered how pursuit is implemented in nature and how this has led to the identification of control laws. In addition, it is evaluated what features these laws exhibit and how they are connected. Secondly, in Chapter 3 pursuit-evasion scenarios are considered from a differential game theory perspective following a formal definition, and it is determined how deep reinforcement learning can and has been used to identify controllers in this setting. Third of all, Chapter 4 considers best practices and recent bio-inspired advancements in neural network architectures applicable to intricate control tasks

such as pursuit. Finally, in Chapter 5 the main findings of this research are summarized, besides an overview of challenges and opportunities as well as recommendations for future research.

2 Strategies for pursuit and interception

Before controllers can be designed with the goal of onboard use in autonomous systems, it is efficient to gather inspiration on the strategies employed by expert natural predators such as dragonfly. Insights into their mathematical formulations as well as arising advantages and limitations can provide the information required to evaluate, implement and improve on these strategies on these strategies. Therefore, this chapter addresses the first of our three sub-questions,

What types of pursuit strategies are exhibited in natural predators, how do they relate, and what associated control laws can be identified?

To properly address this question, this chapter initially formally introduces three types of pursuit strategies and evaluates the identified control laws. After that, these strategies are compared by considering their capabilities with regard to aspects such as time-optimality and motion camouflage. This analysis is continued by considering to what extent the control laws identified can be used to implement multiple and/or hybrid strategies, and evaluate whether and how natural predators such as dragonfly do this. Finally, we hypothesize how a single unifying framework can be used to encapsulate all three strategies and suggest directions for future research.

2.1. Three strategies

A variety of navigational strategies exist for the purpose of interception. This section considers three strategies for predatory pursuit prevalent in nature. To start, their objective is defined alongside the associated cost functions. Thereafter, control laws are identified, which are ultimately compared in a global/general setting. Importantly, this review mainly considers three pursuit strategies that have target interception as their ultimate goal.

Frames, Objectives & Costs

To start, examples for the three strategies are described alongside a visualization in Figure 2.1. First, the simplest form of pursuit is known as *classical*, *smooth* or *pure pursuit* (PP) and is exhibited by certain fly species[3][4]. In this strategy, the pursuer attempts to consistently align its heading with the *line-of-sight* (LoS, λ) to the evader[5]. An alternative to this strategy is formulated by upholding a constant relative heading 'error' rather than perfect alignment, and is exhibited by chasing houseflies[6] and bees[7]. Hence, this second strategy is known as the *deviated*/*biased* pursuit or *constant bearing* strategy[8]. Finally, the third strategy deviates from the previous by striving for a consistent positioning of the evader from the perspective of the pursuer regardless of its motion, thereby achieving a constant target directional vector. Consequently, the strategy is known as *constant absolute target direction* (CATD) and can be observed in hunting falcons [9], dragonflies [10], bats[11] and robber flies [12].

This review is limited to three pursuit strategies as they can be formally combined with a mathematically defined continuous objective ultimately leading to interception. This stands in contrast to other strategies such as *circular* pursuit which merely describes tracking or *saccadic* pursuit which is discontinuous in its objective[5] i.e. the theoretical pursuit definition entails both periods of convergence towards and divergence from the interception target.

The work by Wei et al.[13] is considered which summarizes the associated mathematical cost functions for the three pursuit strategies, following earlier research and derivations by Justh, Krishnaprasad and Reddy[14][15]. These works follow the pursuit perspective of Justh et al.[16] and consider interacting particles moving along curves with constant speed in two dimensions, without further assumptions on pursuer and evader dynamics. This reference frame is known as the *Frenet* frame and should not be interpreted as the body reference frame as it defines attitude/heading with respect to the current curvature. The use of Frenet frames in the aforementioned works ultimately leads to the identification of explicit control laws for the incorporation of the desired strategies. Hence, we opt to follow this perspective/reference frame and the associated variable definition in this review as well, until the identification of said control laws.

The chosen perspective with particles moving along curves as defined by the Frenet frames is visualized in Figure 2.2, wherein x_p and y_p denote the tangential and normal unit vectors for the pursuer's velocity along the current trajectory curvature, respectively. In the Cartesian inertial reference frame these velocity vectors would be conventionally represented as $\mathbf{v}_{\parallel_i}$ and $\mathbf{v}_{\perp_i}$ for agent i respectively. Additionally, the evader's velocity magnitude is described by the scalar ν, governing

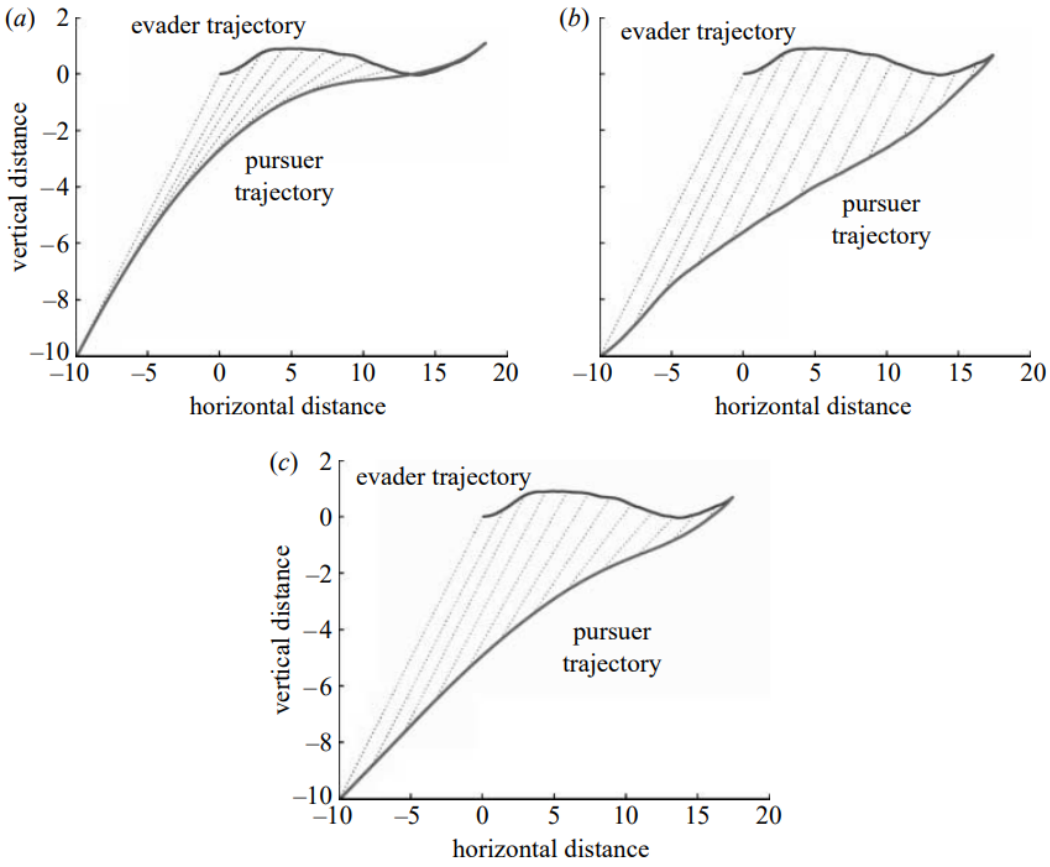


Figure 2.1: Examples of pursuit strategy trajectories for an evader exhibiting random motion in two dimensions. Classical or pure pursuit (a), constant absolute target direction or CATD (b), deviated pursuit/constant bearing (c). Notice that deviated pursuit strategy slightly rotates the LoS based to retain a consistent relative heading of 0.3 rad, while CATD's retains a strictly parallel line-of-sight. Image retrieved from [13].

the relative velocity between pursuer and evader. The variables $\mathbf{r}_p$ and $\mathbf{r}_e$ describe the position vector of the pursuer and evader agents in the Cartesian inertial frame, respectively. Finally, the controller input u controls the agent's heading through control of the curvature's magnitude. Ultimately, this means that the coupled dynamical system in the Frenet frame is represented by Wei et al.[13] as,

$$\begin{aligned} \dot{\boldsymbol{r}}_{\mathrm{p}} &= \boldsymbol{x}_{\mathrm{p}}, & \dot{\boldsymbol{r}}_{\mathrm{e}} &= \nu \boldsymbol{x}_{\mathrm{e}} \\ \dot{\boldsymbol{x}}_{\mathrm{p}} &= \boldsymbol{y}_{\mathrm{p}} u_{\mathrm{p}}, & \dot{\boldsymbol{x}}_{\mathrm{e}} &= \nu \boldsymbol{y}_{\mathrm{e}} u_{\mathrm{e}} \\ \dot{\boldsymbol{y}}_{\mathrm{p}} &= -\boldsymbol{x}_{\mathrm{p}} u_{\mathrm{p}}, & \dot{\boldsymbol{y}}_{\mathrm{e}} &= -\nu \boldsymbol{x}_{\mathrm{e}} u_{\mathrm{e}} \end{aligned} \tag{2.1}$$

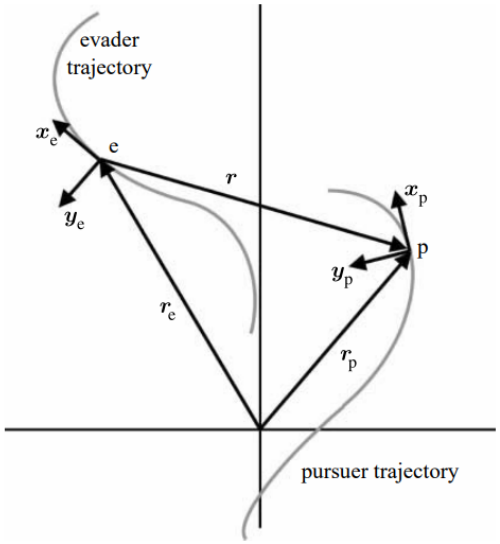


Figure 2.2: Particle reference frames with tangential (**x**) and normal (**y**) velocities for pursuer (p) and evader (e), with relative velocity scalar ν. Image retrieved from [13].

For this perspective, the strategies objectives can be described through cost functions. In this case, a *cost function* describes a mathematical definition that evaluates how the state between pursuer and evader conforms to the desired pursuit strategy in terms of relative orientation, position and velocity. Therefore, this definition of *cost* should not be directly interpreted from an agent's perspective in terms such as time or energy, albeit that logical associations exist. Specifically, the cost variables for the CATD (Γ) and DP ($\Lambda(\theta)$) strategies are respectively represented in Wei et al.[13] as,

$$\Gamma = \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \frac{\dot{\boldsymbol{r}}}{|\dot{\boldsymbol{r}}|} \right) \tag{2.2}$$

and

$$\Lambda(\theta) = \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot R(\theta)\boldsymbol{x}_{\mathsf{p}} \right), \tag{2.3}$$

where the two-dimensional rotation matrix is defined as,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{2.4}$$

with a *deviation* angle $\theta \in (-\pi/2, \pi/2)$, indicating the heading angle deviation. In these equations, **r** defines the range vector between the pursuer and evader respectively, explicitly described by $\boldsymbol{r}_{\mathsf{p}} - \boldsymbol{r}_{\mathsf{e}}$. Notice that in Equation 2.3 if a zero deviation angle is adhered to (i.e. $\theta = 0$), the cost variable $\Lambda(\theta)$ implies that pure pursuit is the optimal behavior. Both cost variables are bounded in range $[-1, 1]$. A negative cost (< 0) indicates reducing range between pursuer and evader. Finally, the associated optimal pursuer and evader geometries are visualized in Figure 2.3, formally referred to as *pursuit manifolds*; describing the set of pursuit states that achieve the minimum cost scores (i.e. -1 scores).

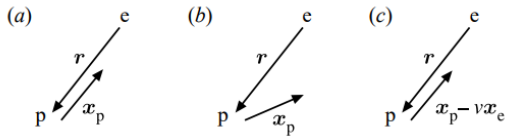


Figure 2.3: Geometric representations of the three pursuit manifolds related to cost variables $\Lambda(\theta)$ and Γ. Cases denote manifolds for (a) pure pursuit, (b) deviated or constant bearing pursuit and (c) motion camouflage pursuit. Image retrieved form [13].

It is important to recognize the cost variables provide a method of scoring pursuit performance according to ideal strategy during pursuit, rather than post-hoc. Moreover, adherence to the objective of a respective strategy (i.e. ideal trajectory) comprises sustained maintenance of a -1 score. For the PP and DP objective (Equation 2.3) this implies that the pursuer's velocity vector is exactly inclined with angle θ with respect to the range vector (with $\theta = 0$ for PP). A score less than -1 indicates that the inclination deviates from the desired θ. For the CATD strategy (Equation 2.2), it implies that the absolute rotation of the range vector does not change, implying that the only relative movement between agents is along the range vector. In this case, deviation from the exact scores equal to -1 or 1 indicate that the range vector does rotate and not all relative movement is along this vector. All in all, the intuition behind striving for a -1 score from the perspective of the pursuer is equivalent for all three strategies; to minimize all relative movement in any other direction than described by the objective.

During pursuit, agents generally do not start on in the desired pursuit states, i.e. on the pursuit manifold. Consequently, these initial stages converging to -1 (i.e. $\{\dot{\Lambda}_\theta, \dot{\Gamma}\} < 0$) are described as pursuer approaching the strategy's *pursuit manifold*[13]. From a game theory perspective, this is interpretable as the pursuer operating in a staging phase, reducing other behavior in favor of acting according to a single *pure* game strategy. On the other hand, any phase before maintained -1 score also implies the contrary; a mixture of strategies/behavior is exhibited.

Control laws

The strategy cost functions define pursuit manifolds, interpretable as the optimal trajectory reducing for the strategy. In turn, closed-loop control laws can be analytically derived which drive the pursuer to these manifolds, where corresponding adherence is scored by -1 of the cost variable. The feedback control laws for pure and deviated pursuit (PP & DP) in two dimensions under the aforementioned assumptions are described by Wei et al.[13] as,

$$\begin{aligned} u_{\mathsf{p}}(\theta) &= -\eta \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot R(\theta)\boldsymbol{y}_{\mathsf{p}} \right) - \frac{1}{|\boldsymbol{r}|} \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \dot{\boldsymbol{r}}^{\perp} \right) \\ &= \eta \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot R(\theta)\boldsymbol{y}_{\mathsf{p}} \right) + \dot{\lambda} \\ &= -\eta\theta_\epsilon + \dot{\theta}_\epsilon \end{aligned} \tag{2.5}$$

where η and $\dot{\boldsymbol{r}}^{\perp}$ indicate the controller gain and the range's perpendicular velocity component from the perspective of the pursuer, respectively. The control law is practically interpretable as a PD controller on an angular error ($\theta_\epsilon = (\theta - \theta_{\mathsf{p}})$),

wherein the former term describes the proportional part with gain (η) and the latter describes the error derivative, equal to the LoS angular rate ($\dot{\lambda}$), with gain equal to 1. The derivative term is introduced to offset ego-rotation, thereby smoothing pursuit[5].

To continue, the feedback control law for the CATD is considered. In an earlier work, Justh et al.[14] derive the feedback control law in two dimensions under the aforementioned assumptions as,

$$\begin{aligned} u_{\mathsf{p}} &= -\mu \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \dot{\boldsymbol{r}}^{\perp} \right) + \left[\frac{(\boldsymbol{x}_{\mathsf{p}} \cdot \boldsymbol{x}_{\mathsf{e}}) - \nu}{1 - \nu\,(\boldsymbol{x}_{\mathsf{p}} \cdot \boldsymbol{x}_{\mathsf{e}})} \right] \nu^2 u_{\mathsf{e}} \\ &\approx -\mu \left(\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \dot{\boldsymbol{r}}^{\perp} \right) \quad \text{(for high } \mu\text{)} \\ &= \mu |\boldsymbol{r}| \dot{\lambda} \\ &= \mu' \dot{\theta}_{\epsilon}, \end{aligned} \tag{2.6}$$

where μ' indicates the *effective* CATD controller gain. In this equation, μ' substitutes for $\mu|\boldsymbol{r}|$ and describes a state-dependent gain adjustment method during pursuit. However, since this state-dependence is also possible through other measures such as the closing speed between pursuer and evader[17], the abstract notation of μ' is preferred.

In their original work on the CATD feedback law Justh et al.[14] acknowledge that Equation 2.6 is not practical, due to intractability of an accurate estimate of the evaders control input, u_e (interpretable as acceleration), in the second term from the perspective of the pursuer. Consequently, they neglect the second term in Equation 2.6 under the assumption that the pursuer opts for a high gain setting in μ', thereby obtaining a practically implementable control law. For this adjusted control law with high gain, convergence to the manifold can still be proven[14]. In addition, under this condition, the controller is practically interpretable as a proportional controller with gain (μ') on the angular error rate or LoS rate.

In subsequent research, the CATD control law is extended, while retaining consistency in the perspectives on system dynamics/definition and its assumptions. The control law is extended for three dimensions by Reddy et al.[15], for stochasticity through sensory noise by Galloway et al.[18], for sensorimotor delay by Reddy et al.[19] and for both sensory noise and sensorimotor delay by Raju et al.[20]. The latter case will be further discussed at the end of this section. For these cases, the derived control laws remain similar to the adjusted control law in Equation 2.6 (i.e. neglecting the second term) in terms of structure and interpretation.

Biological perspective
In comparison, there is resemblance between Equation 2.5 and high-gain Equation 2.6. More importantly, they are deemed biologically plausible as they use requires estimates of range ($\boldsymbol{r}$) or angular error (θ_ϵ) and the perpendicular velocity component ($\dot{\boldsymbol{r}}^{\perp}$) or angular rate variant ($\dot{\theta}_\epsilon$), which can be estimated adequately through evader size and optic flow respectively[21].

Support for the biological plausibility of these control laws can be found in the works of Land et al.[6] and Reddy[22] respectively. In their work, Land et al.[6] identify a PD controller in Equation 2.5 through regression techniques applied to two-dimensional recorded trajectories of houseflies. His study claim success within a $-35 < \theta_\epsilon < 35$ range, even under realistic sensorimotor delay. Similarly, Reddy[22] replicates three-dimensional trajectory recordings of bats, theorized to implement the CATD strategy[11], through the three-dimensional variant[15] of the proportional control law proposed in Equation 2.6 under realistic sensorimotor delay with high correlation.

Control law limits under imperfect conditions
To complete this section, we consider biological and practical insights from the more recent work of Raju et al.[20] on the CATD control law under both sensory noise and sensorimotor delay, following earlier work on either aspect individually[18][19] and works addressing similar questions[23][24]. In this study, Raju et al.[20] use simulation to investigate the robustness of the control law in Equation 2.6 to implement the CATD strategy under a range of delays and sensor noise inversely proportional to the delay setting. They find that for cases with large delays and noise, use of the control law can become undesired as it does not achieve adherence to the CATD pursuit manifold in finite time. This deficiency is further emphasized for increasingly unpredictable/erratic evader movements and larger relative speeds under said conditions. The results by Raju et al.[20] are connected to the trade-off between the required speed of command incorporation and input information accuracy. Consequently, the authors propose that it would be beneficial to formulate an appropriate stopping criterion to break off pursuit in case adherence to the strategy is deemed infeasible (i.e. an *optimal stopping criterion*). It is important to recognize that the work by Raju et al.[20] does not refute the feasibility of CATD through the control law in Equation 2.6 in its entirety, yet highlights limiting conditions and connects these to potential

scenarios encountered during pursuit. Therefore, this work warrants awareness of aforementioned trade-off, underlines the need for noise reduction techniques in this pursuit context and suggests that these limiting conditions should be handled in real-time by decisions such as temporarily switching to another strategy and/or more robust control law variant or alternatively ending the pursuit prematurely.

2.2. Motion camouflage

Sophisticated pursuit strategies are able to reduce visual cues of the pursuer perceptible to the evader; thereby attempting to achieve *motion camouflage* (MC). In motion camouflage, the imminent approach of a predator is practically imperceivable to its prey. For the evader, the pursuer appears stationary at the focal point in its visual field. Note that this situation describes the evader being unaware of the pursuer's motion, yet not necessarily of his presence. Hence, any changes in the relative size of the pursuer as it gets closer might form an alternative perceptible cue of its approach. In insects, this incognisance of the pursuer's motion is due to an inability to leverage information on depth (i.e. range) from optical flow or motion parallax[21], yet even *depth-aware* observers such as humans can lapse and be deceived through these strategies[25]. In other words, during motion camouflage the pursuer achieves a mimicry of the optical flow of the background, thereby making its motion indistinguishable. It is important to realize that this incognisance of approach constitutes an advantage to the pursuer as an evader, oblivious to the ongoing pursuit, which might consequently refrain from adjusting its intended flight path.

For an arbitrary focal point, motion camouflage is achieved by strict pursuer trajectory adherence to the *camouflage constraint line* (CCL) which constitutes the virtual line between this evader's focal point and the pursuer's own position. Given these conditions, Figure 2.4 visualizes three general situations where motion camouflage is achieved. These situations are discussed in the remainder of this section.

Infinite focal point MC

Whenever the prey's focal point is at infinity, motion camouflage can be achieved by maintaining strictly parallel CCL during pursuit, as is visualized in case C of Figure 2.4. Hence, this strategy is also referred to as *parallel navigation*. Practically, an infinite focal point and can be forced upon the prey whenever the predator positions itself against a background void of distinct features, such as in hunting falcons who offset themselves against the sky[9]. The open-loop equations describing the trajectory for parallel navigation have been described by Glendinning[26] as,

$$\boldsymbol{r}_{\mathsf{p}} = \boldsymbol{r}_{\mathsf{e}} + \zeta_t \boldsymbol{F}_{e,\infty} \tag{2.7}$$

where $F_{e,\infty}$ is a fixed unit vector indicating the focal point at infinity of the evader, scaled by a time dependent scalar ζ_t. Together, $\lambda_t F_{e,\infty}$ are interpretable as the irrotational CCL. $\mathbf{r}_p$ and $\mathbf{r}_e$ define the pursuer and evader position vector specifically.

The irrotational objective of parallel navigation aligns with the CATD cost variable Γ. In fact, it can align with the objective of the deviated pursuit/constant bearing strategy ($\Lambda(\theta)$) as well, if the evader (temporarily) moves along a straight line and the pursuer's choice for the deviation angle (θ) counteracts the relative motion of the evader. This describes the condition in Equation 2.3 and visualized in Figure 2.5, where $R(\theta)\boldsymbol{x}_{\mathsf{p}} = \frac{\dot{\boldsymbol{r}}}{|\dot{\boldsymbol{r}}|}$ and equality to Equation 2.2 is observed. Moreover, if the evader retains this straight trajectory, both strategies achieve parallel navigation and time-optimal interception. On the other hand, the relationship between parallel navigation and CATD does not require these conditions on evader trajectory and retains time-optimality even for an erratically moving evader[11].

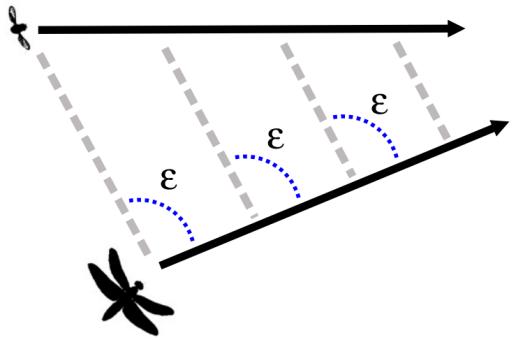
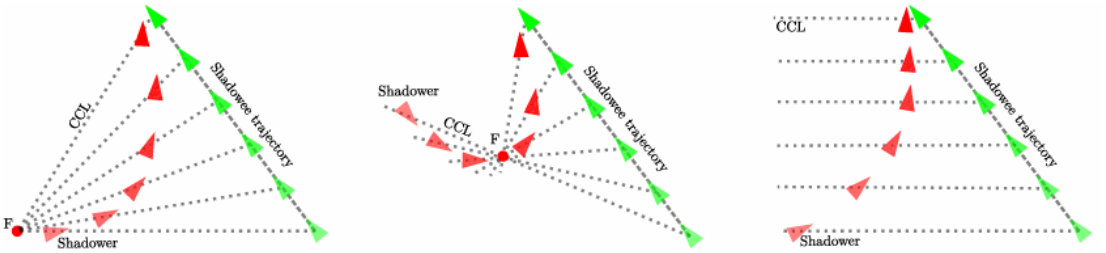


Figure 2.5: Sustained straight evader trajectory and pursuer trajectory with consistently parallel CCL, implying equivalence between constant bearing and CATD pursuit strategies. Image retrieved from [8].

Due to the capacity of CATD in achieving motion camouflage at infinity, Justh et al.[14] define their biologically plausible control law in Equation 2.6 appropriately as the *motion camouflage proportional guidance* (MCPG) law. MCPG is described as *proportional* because Justh et al.[14] acknowledge that the MCPG law resembles *pure proportional navigation guidance* (PPNG) commonly



(a) Motion camouflage pursuit with a focal point behind the shadower

(b) Motion camouflage pursuit with a focal point between the shadower and the shadowee

(c) Motion camouflage pursuit with a focal point at infinity

Figure 2.4: Variants of motion camouflage achieved by the pursuer from the perspective of the evader for different focal point scenarios. Where the first two cases denote a finite focal point, the last case (C) illustrates a focal point at infinity. Image retrieved from [21].

used in missile guidance. The PPNG control law similarly applies acceleration perpendicular to the current direction of motion. This connection is an interesting result by itself, since PPNG follows from a derivation using theory on optimal control with the intent to minimize time-to-intercept and zero-miss-distance [27], rather than reducing missile perceptibility.

Comparable to this connection, the work by Carey et al.[28] further supports the connection of proportional navigation laws to infinite focal point motion camouflage by also identifying the MCPG law, yet seeking it with the objective to minimize energy expenditure and without invoking assumptions on constant speeds. On the other hand, the solution by Carey et al.[28] is not deemed biologically plausible like MCPG, since it requires either prior knowledge on the interception time or absolute initial conditions of both pursuer and evader. Regardless, these aforementioned works indicate the convergence of perspectives towards proportional navigation and its associated benefits, encompassing the robustness to erratic evader motion as well as the minimization of perceivable visual cues, interception time and energy expenditure.

In line with this result, the prominence of CATD is considered from a general evolutionary perspective as well. Besides summarizing pursuit strategies, the aforementioned work by Wei et al.[13] investigates whether a stable equilibrium can exist in a population of individuals implementing pure and deviated pursuit (Equation 2.5) as well as the CATD strategy (Equation 2.6) through the previously defined control laws. Essentially, this experiment involves simulating numerous two-dimensional two-player engagements between the three strategies (with fixed hyperparameters), where achieving an interception faster increases the winning strategy's population probability in the next generation. This simulated game continues until convergence. For this simulation experiment, the triangular probability phase plots for different types of evader dynamics are given in Figure 2.6, where consistent convergence from a diverse set of starting positions towards the CATD strategy (bottom right vertex, implemented through MCPG) can be observed. Hence, this study further underscores the theoretical dominance of CATD, yet is obviously limited to non-reactionary evaders.

Finite focal point MC

In predatory pursuit, assuming an infinite focal point of the evader should not be treated a given. Consider environments with cluttered backgrounds, containing distinct features which can serve as points of reference. In such an environment, detection of the pursuer by an evader means that its focal point is logically set to align with the pursuer's relative position at time of detection and reset for any detection thereafter. Contradictory to the infinite case, this situation implies that the CCL does rotate, which is visualized in Figure 2.4 for correct and wrongful/distracted pursuer positioning by case A and B respectively. For these cases, the open-loop equations describing the trajectory have also been described by Glendinning[26] as,

$$\boldsymbol{r}_{\mathrm{p}} = \boldsymbol{r}_{\mathrm{p}_0} + \kappa \left(\boldsymbol{r}_{\mathrm{e}} - \boldsymbol{F}_{\mathrm{e}} \right), \tag{2.8}$$

with a required initial consistency condition,

$$\boldsymbol{r}_{\mathrm{p}} \times \left(\boldsymbol{r}_{\mathrm{e}_0} - \boldsymbol{F}_{\mathrm{e}} \right) = \boldsymbol{F}_{\mathrm{e}} \times \boldsymbol{r}_{\mathrm{e}_0}, \tag{2.9}$$

where $\boldsymbol{r}_{\mathrm{p}_0}$ and $\boldsymbol{r}_{\mathrm{e}_0}$ denote the initial position of the pursuer and evader, respectively. $\boldsymbol{F}_{\mathrm{e}}$ describes

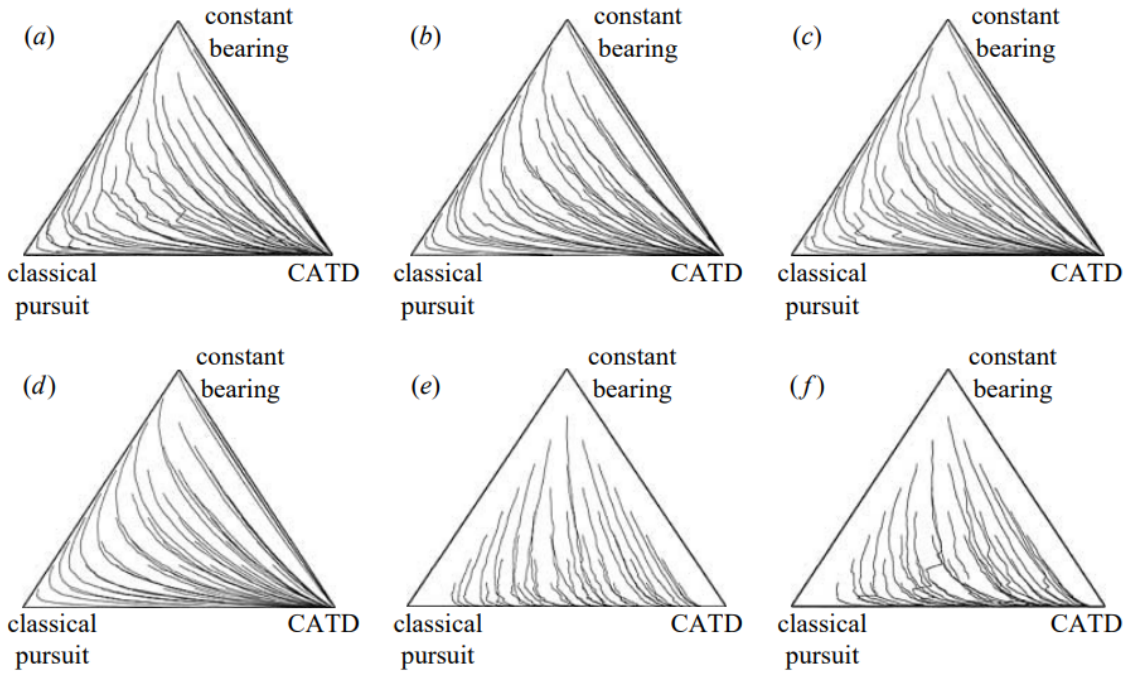


Figure 2.6: Phase plots describing the dynamics of the strategy population probability through a simulated evolutionary game. Notice how the populations start from diverse initial positions, yet consistently converge to the CATD strategy (bottom right vertex), indicating strategy superiority. (a-b) visualize two representative outcomes from simulations with various (stochastic, linear & circular) evader trajectories over 18 trials. (c-d) visualize two representative outcomes from simulations with stochastic evader trajectories over 75 trials. (e-f) visualize two representative outcomes from simulations with circular evader trajectories with random turning rates over 50 trials. Image retrieved from [13].

the finite focal point of the evader. The consistency condition (Equation 2.9) ensures that the initial position of the pursuer in on the CCL, where the remainder of this review solely encounters literature that assumes a focal point aligning with the initial position of the pursuer (i.e. $\boldsymbol{r}_{\mathrm{p}_0} = \boldsymbol{F}_{\mathrm{e}}$) as visualized by case A in Figure 2.4, unless stated otherwise. The scalar $\kappa \in [0, 1]$ defines a monotonically increasing function interpretable as the pursuer's progress towards the evader; initialized at zero and terminating at one at the time of interception.

Besides a formulation, the work by Glendinning[26] establishes the theoretical inferiority of pure pursuit to a strategy achieving motion camouflage at finite focal point for pursuers with consistent constant speed and initial conditions. In his work, the motion camouflage trajectory leads to a faster interception time of an evader following chaotic dynamics. Moreover, he remarks that the assumption of an oblivious evader might not be given and that in the setting that a reactionary evader who is actively attempting to escape likely only further emphasizes these differences between strategies. In fact, Glendinning[26] realizes that motion camouflage offers the possibility of interception even for pursuers with inferior speed, yet for alternative strategies pursuers might have to rely on their superior traits. Logically, these statements extend to infinite case as well.

Contrary to the infinite case, theoretically establishing the connection to motion camouflage at finite focal point to a closed-form control law has proven more complex. This is because the rotational CCL complicates attempts at decoupling the evader's specific motion from the objective of this pursuit scenario, as achieved for the infinite case with Γ in Equation 2.2. In fact, numerous works derive from the pioneering work on motion camouflage trajectory generation by Srinivasan et al. [29], wherein the control problem is described from the perspective of the finite focal point as visualized in Figure 2.7. In this figure, the lateral displacement required ($\Delta\lambda$) to adhere to the CCL is governed by the traversed distance (integrated path, ρ), the distance to the prey as well as the prey's relative rotation ($\Delta\theta$). It is important to recognize that this perspective deviates from the original approach for the infinite case by Justh et al.[14] starting at the objective of pursuit and solely centering around

the pursuer and evader.

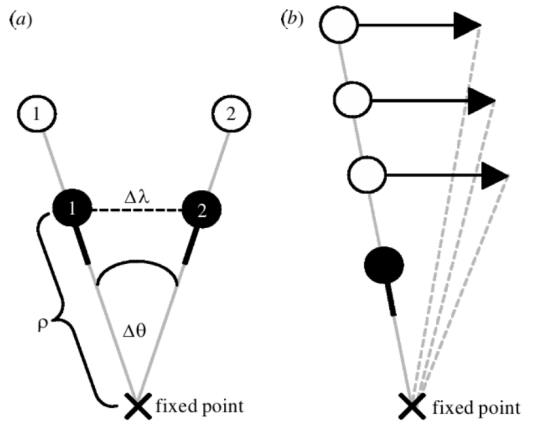


Figure 2.7: Srinivasan et al.'s perspective on finite focal point motion camouflage, where black and white dots indicate positions of the pursuer and evader, respectively. (a) Definitions of fixed point distance (ρ), pursuer's required lateral displacement ($\Delta\lambda$) and evader relative rotation ($\Delta\theta$). (b) Lateral displacement is dependent on the range to the evader. Image retrieved from [29].

For this alternative perspective, a controller capable of achieving motion camouflage with finite focal point was proposed earlier by Anderson et al.[30]. In their work, Anderson et al. define a controller consisting of a three recurrent neural networks to track the traversed distance and predict direction and rotation commands, visualized in Figure 2.8. The controller's subsystems were identified by conventional regression, where ideal input-target pairs for the three subsystems were generated following assumptions with regard to target dynamics such as constant speed. Hence, this estimation method constitutes a form of imitation learning with limited practical guarantees outside the observed state-action space[31].

On the other hand, the controller acquires near perfect motion camouflage trajectories during verification for arbitrary evader trajectories in two- and three-dimensional simulations, as well as validation by analyzing generated pursuit trajectories against two-dimensional recordings of hoverflies. Besides lacking interpretability, the controller claims biological plausibility relying on evidence for path integration in insects[32] and the *availability of inputs*, yet refrains from investigating the impact of sensor noise and/or sensorimotor delay. All in all, this study implies a substantial deviation from the efforts for a unifying framework around proportional navigation.

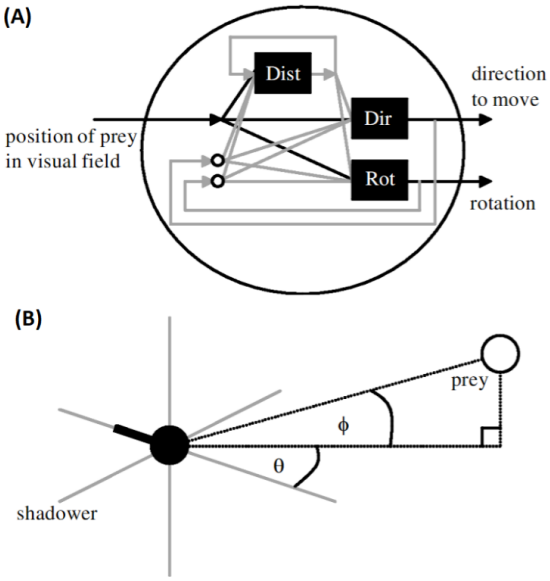


Figure 2.8: Anderson et al.'s controller and visual input overview. (A) The controller consists of three interconnected recurrent subsystems computing the distance to finite focal point (*Dist*) as well as the direction (*Dir*) and rotation (*Rot*) commands). (B) Input azimuth and elevation angles defined as θ and ϕ respectively. Image retrieved from [30].

In contrast to the black-box approach by Anderson et al.[30], recent literature does attempt to establish a direct connection between proportional navigation and motion camouflage at finite focal point. Following results obtained through optimal control theory indicating an approximate linear relationship of relative angular rotations in an earlier study[33], Rano[21] recently identified a closed-form control law for the finite case emerging through policy search. In this work, Rano[21] design as two-dimensional simulation framework to identify a closed-form pursuit controller operating on biologically plausible inputs comprising the relative direction and angles as well as their rates. The reference frame and inputs are visualized in Figure 2.9.

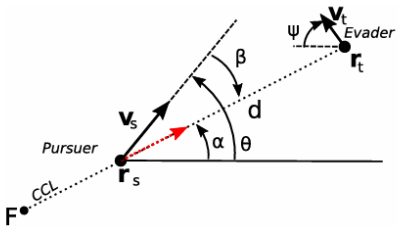


Figure 2.9: Rano's controller input definitions for two-dimensional motion camouflage with finite focal point. $\mathbf{r}_s$, $\mathbf{r}_t$ and **F** indicate the position of the pursuer (*shadower*, subscript s), evader (*target*, subscript t)) and focal point respectively and **v** indicates the velocity. The relative direction, the relative evader angle and the instantaneous angle of the CCL are indicated by d, β, and α respectively. Image retrieved from [21].

Furthermore, this setup assumes constant speeds for both agents and the motion of the evader follows a strictly deterministic straight trajectory, varied with regard to orientation, initial position and relative speed between trials. The reward function employed is visualized in Figure 2.10 and represented as,

$$\begin{aligned} r &= r_1 + r_2 \\ r_1 &= \frac{1}{1 + \left[\frac{\epsilon}{\epsilon_m}\right]^2} \\ \epsilon &= \arccos\left[\frac{(\mathbf{F} - \mathbf{r}_e)\cdot(\mathbf{r}_p - \mathbf{r}_e)}{|\mathbf{F} - \mathbf{r}_e|\,|\mathbf{r}_p - \mathbf{r}_e|}\right] \\ r_2 &= \left[\frac{(\mathbf{v}_p - \mathbf{v}_e)}{|\mathbf{v}_p - \mathbf{v}_e|}\cdot\frac{\mathbf{r}_e - \mathbf{r}_p}{|\mathbf{r}_e - \mathbf{r}_p|}\right]^2 \end{aligned} \tag{2.10}$$

where ϵ is interpretable as the current angular deviation to the CCL from the evader's perspective. $\mathbf{r}_i$ and $\mathbf{v}_i$ denote the position and velocity vector of agent i (either pursuer or evader) respectively.

With regard to the separate reward terms in Figure 2.2, r_1, scores the alignment with the CCL from the perspective of the target, achieving a maximum score in case of perfect alignment implying (retained) motion camouflage. Note that this reward is not a formal and mathematical definition of the pursuit objective for motion camouflage at finite focal point such as Γ and $\Lambda(\theta)$ are for their respective scenarios, yet achieves its maximum when upholding this type of pursuit. The parameter ϵ_m governs the angular *slack* of this reward, where pursuit with $|\epsilon| < |\epsilon_m|$ can be interpreted as motion remaining imperceivable to the evader. Practically, this slack offers a more gradual reward structure and should improve the learning process[31]. The second term, r_2, scores a reduction of distance between the pursuer and evader in an attempt to encourage interception. This reward achieves maximum magnitude for the case where the relative velocity and range vectors completely align and counteract preventing rotation of the range vector, similar to the objective of CATD as described by the cost variable Γ in Equation 2.2.

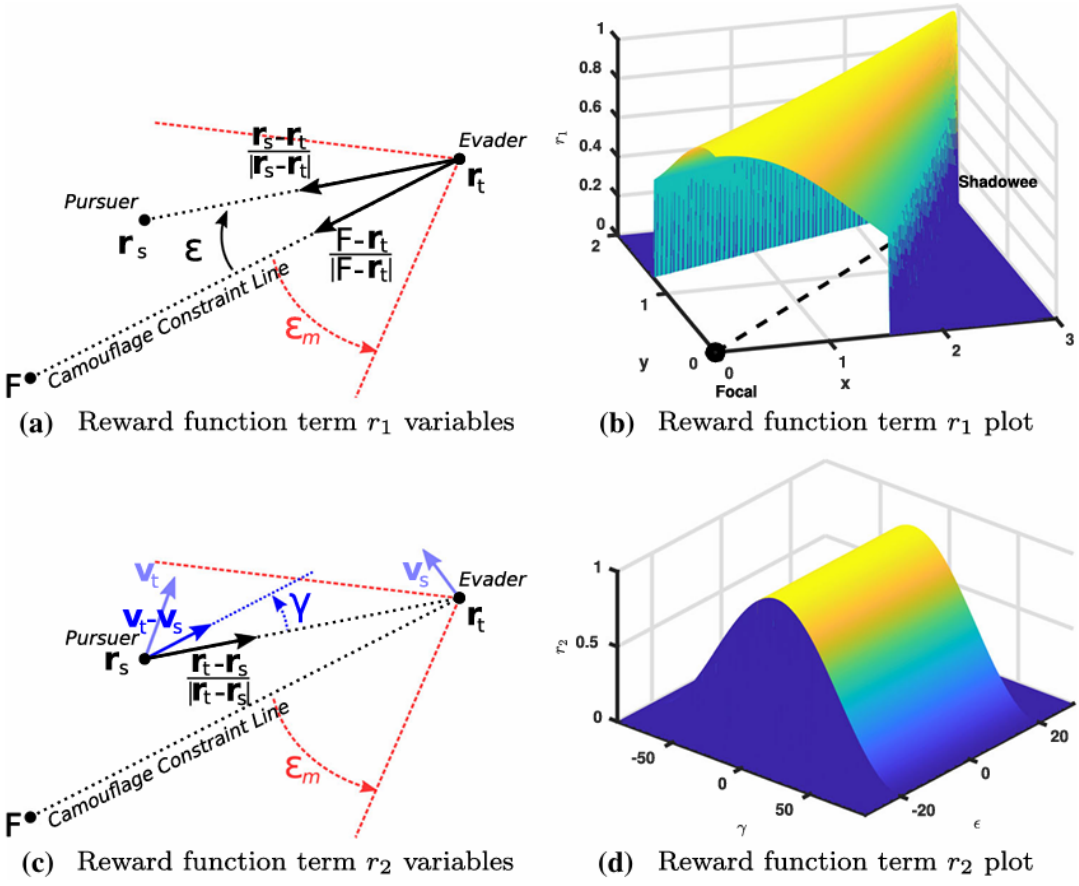
In this study, Rano[21] discovers that a controller based solely on the rotational rate of the CCL ($\dot{\alpha}$) can achieve successful performance on this task. The appropriate gains in this controller are shown to be interpretable as well, being approximately linearly dependent on relative velocity between pursuer and evader. Since the rotational rate of the CCL is attainable by the pursuer (either directly or through $\alpha = \theta - \beta$ in Figure 2.9), this result implies a clear connection to the MCPG control law in Equation 2.6. Therefore, this work suggests a connection between motion camouflage and proportional navigation for a finite focal point, established by Justh et al.[14] for the infinite case. Moreover, this result rejects the need for an estimate of traversed distance (from point **F** in Figure 2.9) obtainable through path integration, advocated in the work by Srinivasan et al.[29] and considered throughout derivative works such as that of Anderson et al.[30].

With regard to linking MCPG to the finite case, support for the perspective can once again be found in the earlier work by Carey et al.[28]. Similar to the infinite case, this work considers control for finite motion camouflage starting from the objective of minimum energy expenditure. Although an impractical open-loop solution is acquired for this case, the authors show that for reactionary evader without constant speed constraints a clear resemblance to the MCPG law can be identified.

On the other hand, in his work Rano[33] acknowledges the limited range of pursuit scenarios considered. Recall that this study considers that evader dynamics follow deterministic straight trajectories and with constant speeds in two dimensions. In contrast to Anderson et al.[30], the sole focus on straight trajectories is a simplification, the use of constant speeds is similar. Hence, it is unknown how this controller holds up in the three-dimensional case. Moreover, one might hypothesize that the simplified evader scenarios itself constitute the reason for the identified controller, the consequent connection to proportional navigation and the rejection of the need of path integration.

Disparity in MC control

Before discussing a unified framework of control



(a) Reward function term r_1 variables (b) Reward function term r_1 plot

(c) Reward function term r_2 variables (d) Reward function term r_2 plot

Figure 2.10: Rano's reward terms utilized for controller identification, displayed in terms of reference frames and reward magnitude. Image retrieved from [21].

laws achieving motion camouflage at finite and infinite focal points, this review further criticizes the link between MCPG and motion camouflage for a finite focal point suggested in the work of Rano[21]. First, reconsider the conclusion associated with Figure 2.5, visualizing the case where a straight evader trajectory implies that both CATD and CB strategies align and achieve motion camouflage at infinite focal point. Secondly, recall that the second reward term, r_2, explicitly resembles Γ (inEquation 2.2) and recognize that its inclusion in Rano's[21] reward definition encourages the identification of controllers implementing the CATD strategy. Finally, recall that r_1 in does not constitute a formal mathematical pursuit objective and acknowledge that its angular slack condition ($|\epsilon| < |\epsilon_m|$) provides substantial reward for a non-stationary focal point. This last situation is visualized in Figure 2.11.

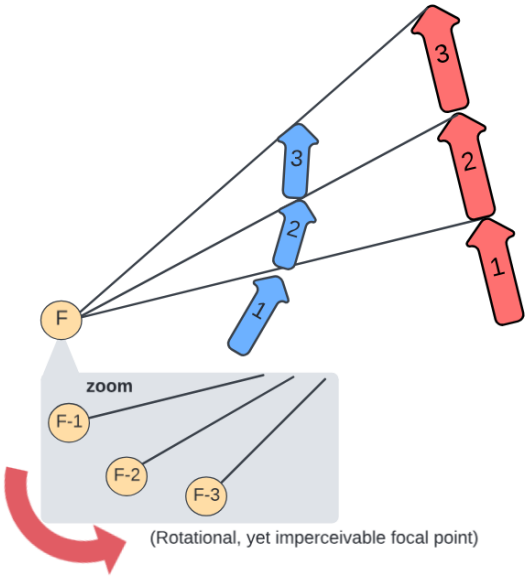


Figure 2.11: Illustration of the (slight) rotation of the finite focal point. Blue and red indicate pursuer and evader trajectory, respectively. It is hypothesized how this small and negligible non-stationarity of the finite focal point becomes imperceivable from the perspective of the evader conditional on it being subject to some angular slack condition, ϵ_m.

By combining these insights, one might hypothesize that the simulation setup defined by Rano[21] encourages the identification of controllers that do not uphold a strictly stationary focal point, yet one that consistently minimizes pursuit trajectory deviation from the CCL within the ($|\epsilon| < |\epsilon_m|$) range. In other words, we hypothesize that in Rano's setup the focal point is not exactly stationary, as motion camouflage at finite focal point would formally require, but is stationary enough for its movement/change to be imperceivable to the evader. Subsequently, the MCPG law constitutes an adequate solution for this setup. Moreover, the exclusive analysis for straight evader trajectories potentially understates the non-stationary nature of the finite focal point. In fact, it is currently unclear whether the adequacy of MCPG in this pursuit scenario is conditional on the evader's trajectory; potentially analogous to the equivalence between the CB and CATD strategies (Figure 2.5) identified earlier. Hence, it is clear that more research is required to theoretically formalize the link between MCPG and motion camouflage for finite focal point unconditionally.

Unifying control laws for pursuit

Although the universality of MCPG cannot be theoretically formalized, the practical use of the MCPG law for motion camouflage is addressed. In practice, the angular slack condition ($|\epsilon| < |\epsilon_m|$) is deemed biologically plausible, as evaders operate with imperfect estimation of pursuer positioning due to sensor noise and delay. This relaxes the need for a strictly non-stationary finite focal point motion camouflage, especially during the initial (farthest) stages of pursuit. Hence, the MCPG law should be able to encompass *practically* motion camouflage at both finite and infinity focal point, albeit that the finite focal point might shift within an imperceivable ($|\epsilon_m|$) range.

Given this insight, the difference between motion camouflage at infinity and finite focal point lies in the choice of gain (μ') for the MCPG law. Where a large gain ($\mu' >> 0$) achieves a *practical* focal point at infinity with irrotational CCL (i.e. parallel LoS), a lower gain setting implies the rotational CCL intersect close to or exactly at some finite point. As the gain reduces further, the shift of finite focal point implied by the rotational CCL becomes perceivable as it falls outside the ($|\epsilon_m|$) range. In fact, recall that equivalence to the deviated pursuit control law (Equation 2.5) arises for a gain equal to one. Thus, a gain spectrum for the MCPG law can be identified, with the CATD strategy on one end and deviated pursuit at the other. In between these points, the MCPG law achieves *practical* motion camouflage for a finite point. This hypothesis is visualized in Figure 2.12 and describes the *practical* unification of MCPG across pursuit strategies.

This practical unifying perspective finds support in the study by Fabian et al.[17]. In their work, Fabian et al.[17] study the interception trajectories of two killer fly species that vary in hunting environments and ranges, achieving motion camouflage at infinity and finite focal point. Their study finds that in both cases, the MCPG law can replicate the interception trajectories for both linearly and erratic moving evaders adequately in terms of rotational errors. In contrast, they also show that this is not true for controllers implementing the deviated or pure pursuit strategies and find that proportional navigation is consistently more efficient than the pure and deviated pursuit strategies in terms of interception time, underscoring this mathematical aspect[26] empirically.

Furthermore, Fabian et al.[17] find empirical support for the observation by Rano et al.[21] that the interception efficacy of the MCPG law is subject to the relative velocity between pursuer and evader and that this is encoded in the gain. In another work, Wiederman et al.[34] further observe that sensory neurons in dragonflies are capable of formulating and utilizing a prior expectation of the evader's relative position in the visual field during pursuit. Subsequently, the incorporation of information into the effective gain implies it might encode a predictive component as well. Although these statements might diverge on the exact information encoded in the effective gain (μ' in Equation 2.6) of the MCPG law, they serve as support for its hypothesized state dependence.

The hypothesized state dependence can also be considered from a practical perspective by reconsidering the comments on the MCPG limits under imperfect conditions (i.e. delay & noise) discussed at the end of Section 2.3. There, it is discussed how the work by Raju et al.[20] identifies how the MCPG law cannot implement the CATD strategy in certain limiting scenarios. Therefore, these cases require decision-making in real-time to adjust strategy or stop pursuit entirely. Assuming that continued pursuit is desired, this suggests the pursuer will have to implement a (temporary) divergence of intended strategy, achievable by adapting the gain setting. An example can be formulated by reconsidering the gain spectrum inFigure 2.12, where this change in strategy from CATD to pure pursuit would be implemented by a reduction in gain μ (i.e. leftwards on the spectrum). Once again, the fact that the flight condition would trigger this adaption

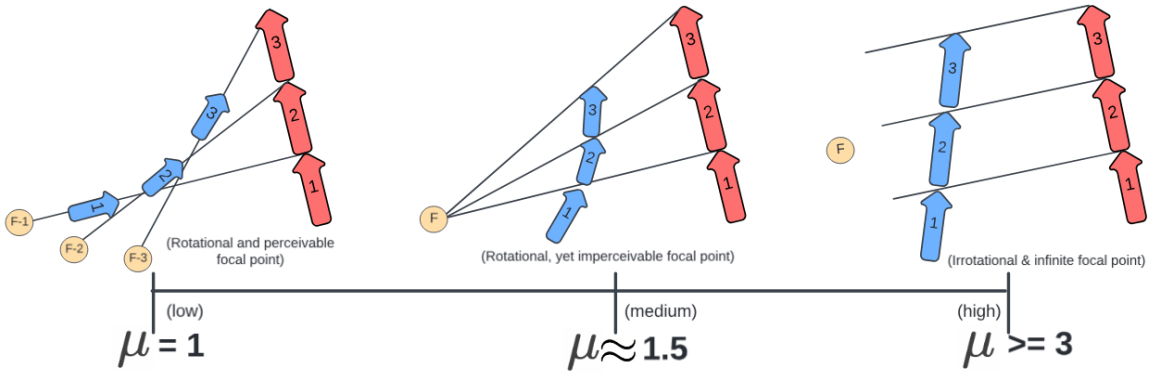


Figure 2.12: Illustration defining the hypothesized relationship between effective gain setting μ' in the CATD control law (Equation 2.6) and pure pursuit (left) and finite (center) and infinite (right) motion camouflage trajectories respectively. Blue and red indicate pursuer and evader trajectory, respectively. The hypothesis defines that for unity gain of μ' in the CATD controller, pure pursuit is implemented with clearly perceivable rotation of the CCL (i.e. no motion camouflage). For higher gains, the rotation of the CCL is hypothesized to reduce. This ultimately results in finite focal point motion camouflage (center), assuming that the small rotational rate of the focal point is imperceivable from the perspective of the evader (see close up in Figure 2.11). For even larger gain settings, the CCL becomes truly stationary and infinite focal point motion camouflage (right) is achieved.

implies a state-dependence implementation of the MCPG law during pursuit.

All in all, this section has discussed the utility of the MCPG law, originally formulated by Justh et al.[14], in acquiring motion camouflaged pursuit trajectories. Most importantly, we hypothesized that differences across strategies (e.g. DP and CATD) as well as pursuit conditions (e.g. relative velocity) can be explained through a change in gain setting for the MCPG law. Specifically, it identified how the choice of gain in the MCPG law comprises a spectrum bounded by the deviated and CATD pursuit strategy. In addition, it discusses how the choice of gain is likely state-dependent in insects, encoding fundamental information on pursuit conditions encountered during the scenario. Consequently, these aspects emphasize that the design of artificial and competitive pursuit controllers derived from MCPG law might leverage from careful consideration of dynamic state-dependent gain management.

2.3. Mixing MCPG to ensure interception

Considering the established prevalence of the MCPG law amongst theoretical strategies, it is further considered how well it explains hunting behavior in expert predators. To this end, this section considers the oldest airborne predator known to achieve motion camouflage, the dragonfly[10]. This predator has been popular throughout research due to the fact that they implement their sophisticated pursuit with superb interception capabilities, reaching interception efficacy of up to 95%[8]. Furthermore, it considers how changing strategy during pursuit can improve interception chance.

MCPG and dragonfly

A biologically plausible reproduction of the dragonfly sensorimotor system for the purpose of target interception has recently been implemented by Plunkett and Chance[35] in a three-dimensional simulation setting, following an earlier model by Chance[36]. This replication considers a two-dimensional *prey-image* and the fovea's relative position as acquired by the dragonfly's perception system. Subsequently, a simplified replication of the dragonfly's nervous system uses this information to acquire a mapping to separate rotational commands of the head and body. In simulation setting, their model is successfully able to achieve interception for simple target trajectories, with a strategy resembling proportional navigation. The authors note that this resemblance arises through the attempted cancellation of *prey-image slippage*, defined as the relative in motion of the prey in the input image with respect to the fovea and is visualized in Figure 2.13. Despite the simplifications, the work raises practical support to address whether dragonflies are capable of implementing the MCPG guidance law for pursuit and interception at all.

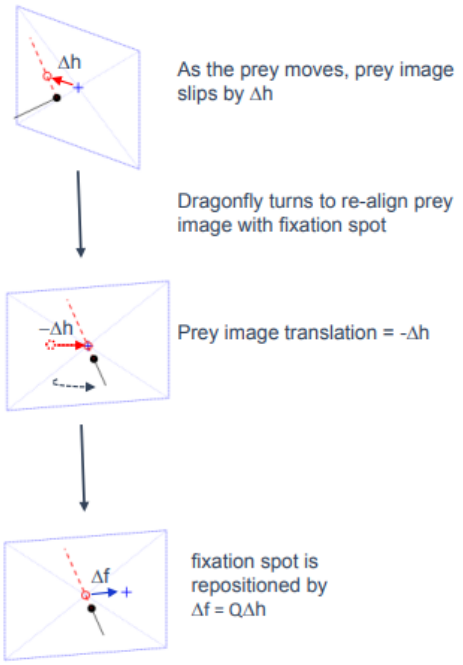


Figure 2.13: Sequential overview of replicated dragonfly control strategy through attempted cancellation of *prey-image slippage* with respect to the fixation spot/fovea by Plunkett and Chance[35][36]. Image retrieved from `https://www.osti.gov/servlets/purl/1894019`.

With regard to the recording and analyzing the hunting behavior of actual dragonflies, the work by Mischiati et al.[37] is fundamental through its proposed descriptions of the dragonfly's control system during interception. This study is fascinating due to precise recording of the insect's behavior during interception of the fruitfly and artificial prey (beads along linear trajectory). This behavior is recorded both on trajectory level as well as separate head and body orientation through the use of mounted sensors.

In their work, Mischiati et al.[37] similarly identify *prey-image slippage* as an informative error signal yet advocate that the dragonfly interception is too accurate and complex to be merely reactionary and driven by a fixed feedback controller implementing the MCPG strategy. Support for this claim is based on empirically observed deviation from the CATD objective (i.e. $\Gamma = -1$, Equation 2.2) during interceptive flights, visualized in Figure 2.14. Notice from this image that while on average convergence towards the manifold can be observed, individual flights present substantial deviations. These observations indicate consistent trajectory misalignment with regard to the infinite focal point, i.e. a rotational CCL. Furthermore, they note a clear disassociation between dragonfly steering events and prey steering events (and vice versa), one might expect in conventional feedback control mechanisms.

As an alternative to a feedback controller, Mischiati et al.[37] propose an intricate decoupled system for head and body control to combat *prey-image drift/slippage* and retain prey tracking on the fovea.. Specifically, they hypothesize that in order to meet this objective both a forward model with efferent copy of body motion and at least a simple predictive model of relative prey translation is required, such as one assuming constant prey speed. Where inclusion of the former is argued to be essential based upon the substantial prey-image drift invoked by body motion, the latter is supported by observing (residual) head movement occurs with delays superior to perception.

In opposition to this perspective, the aforementioned work of Fabian et al.[17] raises two arguments based on their study on killerfly. First, they argue that the incorporation of the MCPG law does not exclude the simultaneous existence of more intricate models responsible for e.g. tracking ego-rotation and/or evader prediction. In turn, this aligns with insights from Section 2.2 describing state-dependent gain encoding, as this additional information can be encoded into MCPG law as well. Secondly, they note that the consistent deviation from $\Gamma = -1$ might indicate an imperfect attempt at the CATD objective, rather than determined divergence from it. This argument originates from the accuracy in killerfly trajectory replication by MCPG and might hold for dragonfly as well by considering Figure 2.14 and recognizing that therein the mean $\dot{\Gamma}$ (i.e. change in Γ) shows sustained periods of negative magnitude, interpretable as convergence towards the CATD pursuit manifold. All in all, these remarks do not reject the practical universality of MCPG implied by Mischiati et al.[37].

Mixing strategies in practice

Although the mean $\dot{\Gamma} < 0$ is interpretable as general intent at CATD across flights, Figure 2.14 contains numerous flight examples with (lengthy) periods of $\dot{\Gamma} > 0$. In hunting fruitfly (f in Figure 2.14), this might be explained a repositioning following by erratic evasive motion. On the other hand, this explanation does not explain the pursuit trajectory for artificial prey following linear trajectories (g in Figure 2.14). An alternative hypothesis constitutes that dragonfly dynamically switch strategy during pursuit, where this strategy state is referred to as a *mixture of strategies* as it deviates from utilizing

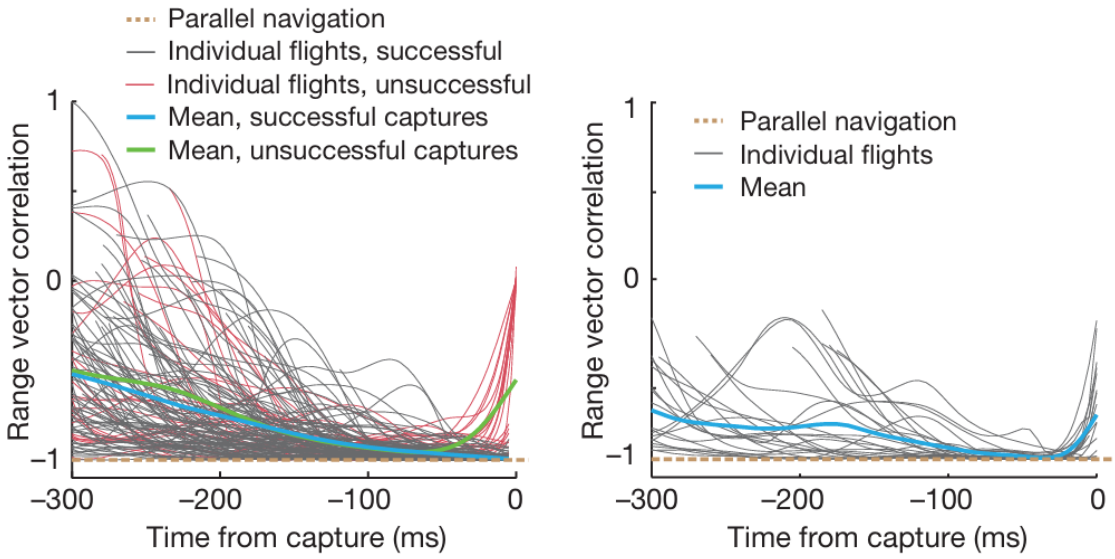


Figure 2.14: Empirical trajectory Γ (CATD objective, *range vector correlation*) as a function of *time to capture* for dragonfly hunting fruitfly (f) and artificial prey (g) in experimental setting respectively. Image retrieved from [37]. Recall that *parallel navigation* refers to the case where the CATD strategy (Equation 2.6) achieves optimal trajectory, i.e. whenever $\Gamma = -1$ (Equation 2.2).

either of the three *pure* strategies. The practical universality of MCPG across strategies addressed in Section 2.2 suggests that this feat is accessible through adjustment of current effective gain (μ'). Hence, the trajectory observations in Figure 2.14 might serve as empirical evidence for the practical link between pursuit strategies established by MCPG, as well as the subsequent fluent access between strategies it implies.

The recognition of a dynamic mixture of strategies in pursuit behavior seemingly stands in contrast to the original perspective discussed in Section 2.3 of this review, outlining (ideal) control laws to converge to *pursuit manifolds* according to a desired type of approach. However, one might reconcile mixing as a form of robust behavior by considering the potential practical difficulties encountered during pursuit and addressing the overall objective at hand; to ensure interception itself. Besides sensory noise and delay, one can imagine numerous practical difficulties arise when considering reactionary evaders actively attempting escape. Although the identification of mixtures in dragonfly has not been established, mixing pursuit strategy behavior has been identified in other organisms such as blowfly[38], fish[39] and humans[40].

All in all, this section has reviewed literature on dragonfly hunting behavior. Although discrepancies to the theoretical results implied by MCPG control law are apparent in recorded trajectories, its presence cannot be directly rejected. Moreover, the universality of MCPG is recognized to allow for fluent access between strategies even during pursuit. Consequently, we formulated a hypothesis for an implemented mixture of strategies to potentially explain the empirical behavior observed in dragonfly, yet subsequent research for this perspective is required. Practically, mixing of strategies provide yet another tool to overcome erratic pursuit encounters and improve robustness with regard to interception.

2.4. Final remarks

This chapter has considered interceptive pursuit and introduced the pure, deviated and CATD strategies according to their respective objectives as well as formulating control laws. Amongst these control laws, CATD can be implemented according to the *motion camouflage proportional guidance* (MCPG) law. This control law with state-dependent gain is attributed beneficial traits including the robustness to erratic evader motion as well as the minimization of perceivable visual cues, interception time and energy expenditure. Furthermore, this chapter has established how adjustments in the MCPG gain can encapsulate several pursuit scenarios such as motion camouflage at infinity and finite focal point as well as alternative strategies such as deviated pursuit and varying pursuit conditions.

The prevalence of MCPG across strategies is supported empirically in nature for certain insects, and

it is hypothesized how expert hunters such as dragonfly might utilize the state-dependent gain modulation in MCPG to mix strategies, observed in other species, in an attempt to improve their chance of interception. Consequently, this hypothesis can inspire the design of competitive onboard pursuit controllers intended for robust interception through attentive and dynamic state-dependent gain management of the MCPG law. A literature gap exists for this perspective currently, where one could research how the MCPG law with state-dependent gain dynamics should be implemented and/or how it arises in practice, onboard control systems capable of attaining robust and consistent interception of reactionary evaders. In order to conclude on these question, more research from a control theory perspective is required.

3 Games of pursuit and evasion

Throughout the considered literature a focus on pursuer dynamics has been prioritized, where evader dynamics are subject to simplifications, limiting constraints or completely neglected[21][36][35]. However, the design of robust control systems capable of pursuit and interception warrants the considerations of a perceptive adversary with reactionary evasive dynamics. This perspective gives rise to the avenue of pursuit-evasion scenarios, wherein the interception task is considered a continuous (non-sequential) game between two agents/teams with adversarial objectives, formally described through *multi-agent pursuit-evasion differential game theory*.

Consideration of controller design through differential games can provide robust systems through the optimization of criteria against the worst potential actions of an adversary[41]. Consequently, this chapter addresses the following sub-question,

How can differential game theory be used to improve the robustness of pursuit controllers attempting interception of reactionary evaders?

To properly address this question, this chapter initially provides a formal definition of differential games of pursuit and evasion, as well as considering optimal solutions. Afterwards, it considers general yet relevant theory as well as current best practices on how to apply deep reinforcement learning in order to approximate these solutions for multi-agent game scenarios. Finally, it identifies practical implementations of deep reinforcement learning and differential game theory for pursuit evasion scenarios and attempts to identify best practices to identify robust controllers in this setting.

3.1. Differential games

Differential games define a system representation of a dynamical conflict wherein a differential equation governs the evolution of the game as players (agents) influence the game's state and payoff through their actions[41]. Differential games can be used to formulate a wide variety of interaction types between agents and game scenarios. The differential game of pursuit-evasion in this review is restricted to two agents, one pursuer and one evader. Specifically, in this interception scenario, the pursuit-evasion task comprises a zero-sum game with time-to-intercept as payoff/cost[42]. Formally, this scenario comprises a differential *game state* equation (f) as well as the *performance functional*(J) and Value function (V) defined as,

$$\dot{\mathbf{x}} = f(t, \mathbf{x}(t), u(t), v(t)) \tag{3.1}$$

$$\begin{aligned} J(\mathbf{x}(t), u(t), v(t), t_0, t_f) &= q\left(\mathbf{x}\left(t_f\right)\right) \\ &\quad + \int_{t_0}^{t_f} g(t, \mathbf{x}(t), u(t), v(t)) dt \\ V(\mathbf{x}(t_0), t_0, t_f) &= \min_{u(t)} \max_{v(t)} J(\mathbf{x}(t), u(t), v(t), t_0, t_f) \end{aligned} \tag{3.2}$$

for $t \in [0, t_f]$ and $\mathbf{x}(0) = \mathbf{x}_0$[42]. Where in these definitions $\mathbf{x}(t)$ defines the system/game state and $u(t) = \mu(t, \mathbf{x}(t))$ and $v(t) = \nu(t, \mathbf{x}(t))$ describe the pursuer's and evader's arbitrary feedback control policies under perfect observation of the complete state, respectively. In Equation 3.2, functions g and q define the running- and terminal cost, respectively. Importantly, the difference between J and V lies in the fact that the performance functional scores any game state and the value function provides the optimal value of J subject to the optimal controls of both agents. Note that in this definition the $minmax$ expression holds in reverse as well, since the game progresses in a continuous manner.

Now, for this system and performance functional definition, an equilibrium is described by the control inputs (superscript *) conditional on the state that satisfies the *Hamilton-Jacobi-Isaac* (HJI) partial differential equation (PDE),

$$-\frac{\partial V}{\partial t} = \frac{\partial V}{\partial \mathbf{x}} \cdot f\left(t, \mathbf{x}, u^*, v^*\right) + g\left(t, \mathbf{x}, u^*, v^*\right) \tag{3.3}$$

with $V(t_f, \mathbf{x}) = q(\mathbf{x})$[43]. Where the functions V, g define the value and running cost function, respectively. f defines the game state equation.

These previous equations and conditions generalize the *Hamilton-Jacobi-Bellman* (HJB) PDE formulation of the optimization objective and its dynamics used in *optimal control* theory from a single agent to multiple ones. In that single agent setting, the optimal control input then minimizes the performance functional with regard to $u(t)$ e.g. minimization of controller effort and of deviation from key states in LQR[44]. In this optimal control setting, any other agent's state and input ($v(t)$ here) is assumed to be part of the *environment* and omitted from the formulation. The main agent's control $u(t)$ input is optimized with regard to this environment definition, without consideration of the potentially adversarial, objective of any other agent. This stands in contrast to the multi-agent differential pursuit-evasion game, wherein the zero-sum setting implies the contrasting minimax objective for the performance functional in Equation 3.2 between the agents. Hence, the equilibrium state for this minimax objective is described as a saddle-point equilibrium or *Nash* equilibrium.

For controller design and associated parameter optimization, the Nash equilibrium is targeted as it provides a stable optimum, robust to perturbations and minimizes worst-case losses[42]. It is interpretable as the game state where no single agent should adjust his policy unilaterally, as no improvement can be made in consideration of the other agent's strategic response. Consequently, it is geometrically defined as a saddle point; describing the point at which the two contrasting objective curves meet. Subsequently, the task at hand is to acquire a strategy that attains this Nash equilibrium by adequately determining the value function V (Equation 3.2) and associated optimal control policies u^* and v^*, extracted through $argmin$ and $argmax$ rather than min and max in Equation 3.2 respectively.

Challenging differential game scenarios
In practice, the pursuit-evasion games proceed under particular conditions with respect to agents and environment, warranting pursuit-evasion game revisions through reformulations of the associated game differentials and performance functional. , summarized here. To start, alternative game differentials can be used to analyze the impact of agent asymmetry or heterogeneity under constraints on dynamics (e.g. speed/maneuverability)[45][46] as well as partial observably (range of sight) [47][48]. Furthermore, environments might include obstacles[49][50] or specified regions of interest[51], further complicated through asymmetric agent restrictions. Moreover, analysis has been performed on stochastic controller dynamics[52] as well as imperfect information provision through stochastic or incomplete observations of the game state[53][54], once again complicated through agent asymmetry to these aspects[55][56][57].

Games can also consider specific realistic scenarios, such as for aerial engagements. For this specific three-dimensional aerial scenario, analysis has been conducted into fighter dogfights[58], but also heterogeneous agents in the form of missile versus fighter/aircraft scenarios[59][60][61]. Finally, fixed agent dynamics can be relaxed by considering a finite number of alterations to agent dynamics during a game for the pursuer[62] or evader[63].

3.2. Deep Reinforcement Learning methods

Although the differential game representation of the pursuit-evasion scenario is insightful and the HJI PDE condition can prove optimality of an obtained strategy, extracting such strategies analytically can prove complicated or intractable. This inherent complexity only increases when further realism is desired and the game considers additional characteristics of the environment and its agents, as outlined at the end of the previous section. In general, this analytical intractability holds for cases with non-linear dynamics and constraints[42].

Recent decades have seen significant advancements in the approximation of strategies implied by the HJI PDE condition on a wide array of tasks through combinations of novel *deep learning*[64] architectures and *reinforcement* algorithms[31] evolving into the field of *deep reinforcement learning* (DRL). Specifically, this application to differential games is achieved by utilizing DRL to approximate the Value function ($V(\cdot)$ in Equation 3.2) or the agents' optimal actions ($v(t)$ and $u(t)$) or both. In the remainder of this section, the implementation of DRL discussed, with a focus on design paradigms relevant for differential games of pursuit and evasion. Before addressing multi-agent games, single-agent setups are considered. This is because throughout relevant literature one can observe that advancements in and understanding of DRL is recognized to frequently develop in this setting, with subsequent conversion to multi-agent scenarios. In consideration of this pattern, succeeding paragraphs address single-agent DRL algorithms, their characteristics as well as best practices before reviewing their translation to the multi-agent setting. The section is completed with a comment related to alternative approximation

methods.

DRL for single-agent scenarios
Inception of DRL can be attributed to the *Deep Q network* (DQN) algorithm by Mnih et al.[65], combining deep learning and Q-learning[66] for single-agent strategies with discrete action spaces. Subsequently, *Deep Deterministic Policy Gradient* (DDPG) by Silver et al.[67] bridges the gap for single-agent strategies involving continuous action spaces by establishing a similar connection to policy gradient theory. Finally, in another work, Mnih et al.[68] define the *Asynchronous Advantage Actor Critic* (A3C) and synchronous *Advantage Actor Critic* (A2C) which comprise alternative methods for single-agent settings in both discrete and continuous action spaces.

In contrast to the Bellman equation's Q-value definition, the *advantage* value definition scores deviation of a state-action pair's value to the state's average value (irrespective of the selected action) in an attempt to reduce variance experienced during estimation. In direct contrast stands the *regret* value definition, which is the to-be minimized value of the current policy compared to the best policy[69].

All of these methods define (at least) two dedicated networks named the actor and critic, responsible for policy and (Q-)value estimation respectively. In this formulation, the critic (value network) evaluates (future) states, which in turn guides the policy network towards the optimal strategy during the estimation phase[31]. Furthermore, for both on- and off-policy methods an *experience replay buffer*, containing representative state-action pairs acquired throughout the estimation phase, can be maintained to smooth improvement and promote convergence during estimation and combat phenomena such as catastrophic forgetting[31]. After estimation, the critic is dropped and the policy network (often denoted $\pi_\theta(s, a)$) is retained as it comprises the agent's strategy state-action mapping, i.e. the controller.

Amongst DRL methods for single-agents, an improved version of A2C[70] known as clipped *Proximal Policy Optimization* (PPO-clip) proposed by [71] is prominently used in research currently. This is attributed to its favorable trade-off between method simplicity, computational efficiency, estimation stability as well as adequate sample efficiency [71][72]. On the other hand, disadvantages include that PPO produces conservative gradients which lengthen the learning process in cases where optimal and current policy are distant, it introduces more hyperparameters (showing potential sensitivity to their choice) and its sampling efficiency is exclusive to environments with cheap sampling.

Amongst single-agent PEG research, with either agent using an optimized model but not both, the PPO[71] and DDPG[67] algorithms feature most prominently as summarized in Table 6.1. Where both algorithms rely on policy gradient theory, their fundamental difference lies in their on- and off-policy characteristic respectively. The off-policy DDPG with higher sample efficiency has increased memory requirements as well as implementation complexity through the use of its replay buffer, which is thought to stabilize convergence[67]. On the other hand, the on-policy PPO with lower sample efficiency is easier to use as it does away with this buffer, with successful implementations for single-agent PEG scenarios research[73][74][75]. Moreover, it can be argued that pursuit-evasion games do not require the high sample efficiency of DDPG, as the simulated environments are generally cheap with regard to sampling. All in all, it can be observed from Table 6.1 that for this single-agent PEG setting, research utilizing PPO utilizing or DDPG is split. Further arguments highlighting the perks of either method in general and multi-agent PEG setting are discussed in the next section.

DRL for multi-agent scenarios
DRL has been applied to a wide variety of multi-agent games, yet this review is mainly restricted to adversarial two agent scenarios. Applications of such DRL approaches to differential games is not as straightforward as introducing a multitude of agents with the appropriate individual networks, and requires careful consideration of the aspects a multi-agent scenario implies.

In a survey, Hernandez et al.[76] summarize fundamental differences of multi-agent to the single-agent scenarios identified throughout relevant literature. These include the presence of multiple agents, invoking non-stationarity of the environment/game state from the perspective of individual agents. In addition, the curse of dimensionality, describing the increasingly sparse relationship models controllers have to identify, is governed by the number of agents in the game as they increase the size of the state-space. Furthermore, it remains challenging to attribute reward for individual agents' contribution to the game. Finally, agents might exhibit limited generalization whenever optimization get stuck in local optima and agents' fail against alternative strategies. Related to this, the use of experience replays (i.e. *replay buffer*) to improve convergence time in single-agent scenarios[31], requires careful consideration in multi-

agent scenarios because the replays might get outdated at a faster pace causing an agent to (mainly) train against an outdated adversary's policy.

To limit these challenges, the Hernandez et al.[76] also identified best practices throughout literature. First, the replay buffer is identified as worthwhile, yet it is appended by additional information describing sample characteristics. It can be interpreted as a *history of strategies* rather than history of experiences and can be used to determine relevancy of samples at the current stage in estimation phase. This aspect also relates to potential improvements with regard to performance generalization, where an alternative approach is to introduce ensembles in policy networks capable of embedding multiple (conflicting) strategies into the agent.

Furthermore, the *centralized training with decentralized execution* (CTDE) principle is identified as combating non-stationarity by conducting state-action evaluation during the estimation phase from a centralized perspective rather than at individual agent level[76][77]. Subsequently, this principle is empirically rather than theoretically accredited with improvements in estimation speed, stability and variance as well as agent robustness and performance[78]. Implementation is achieved by defining centralized critics operating on the complete rather than local/individual game state, visualized in Figure 3.1. Execution of policy (i.e. state to action) remains decentralized at the individual agent level. Drawback of this principle is the increase in the number and the size of networks (critics/actors) required, compared to a completely decentralized approach. This scale-up results from increased state-spaces and the introduction of additional cross-agent approximation networks required for updating during estimation phase[77].

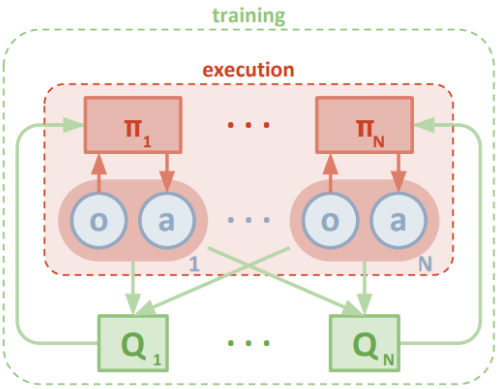


Figure 3.1: Multi-agent scenario with centralized training and decentralized execution principle in actor-critic (π_i & Q_i respectively) system setup. o_i indicates the (partial) game state observed by an individual agent, while a_i indicates the agent's action. Image retrieved for MADDPG architecture from [77].

In light of these drawbacks and the lack of theoretical guarantees, Lyu et al.[78] investigate CTDE versus the decentralized perspective. Their work highlights that both perspectives acquire the same expected parameter gradients. In addition, CTDE is theoretically proven to update decentralized actors with more variance. Hence, the authors hypothesize that improved critic's value-function estimation stability does not translate to the reduced variance in the actors' policy gradients[78] and present empirical support. On the other hand, they recognize that decentralized critics potentially encounter higher bias due to limitation in samples and inferior value function estimates. All in all, the authors conclude that a bias-variance trade-off arises in practice, with neither perspective being strictly superior[78].

To continue, another best practice is observed through the use of recurrent neural networks. These network types have been identified as powerful tools in MADRL due to their inherent design focusing on high network expressiveness and dynamic access to memory depth, enabling them to learn complex spatial-temporal dynamics observed in differential games[76].

In sight of these best practices, various single-agent DRL algorithms have been adapted for multi-agent scenarios, beside dedicated multi-agent approaches. The *Multi-agent deep deterministic policy gradient* (MADDPG) algorithm by Lowe et al.[77] expands on DDPG by implementing a previously identified best practice and is introduced by considering centralized critic networks and decen-

tralized actor networks operating on global and local game state respectively, visualized in Figure 3.1. The same technique can be applied to the A2C, A3C, DQN and PPO methods discussed previously to formulate their multi-agent counterparts.

Amongst these multi-agent counterparts, off-policy (e.g. MADDPG) rather than on-policy methods (e.g. PPO, A2C) are featured more prominently in MADRL and show state-of-the-art performance on multi-agent benchmark tasks[79]. This deviation from single-agent scenarios is mainly attributed to the higher sample efficiency combined with the effectiveness of replay buffers in these increasingly complex environments[80]. Amongst these off-policy methods, the class of MADDPG[77] networks remain conventional choices throughout relevant literature[80], albeit that researchers have derived enhanced versions for dedicated tasks.

On the other hand, it should be noted that the use of centralized methods as well as the proposed off-policy algorithms scale worse than their alternatives with respect to network size and amount. Specifically, this scale-up can be attributed to two main components; the approximation of other agent's policies from the perspective of an agent and the use of *soft* architectures. Besides often requiring larger networks for multi-agent environments[76], centralized and off-policy multi-agent methods such as MADDPG[77] maintain a separate approximation network for every other agent's policies from the perspective of a single agent in order to properly implement the centralized critic. Moreover, soft architectures imply double the amount of networks is used; a *running* and *target* network are utilized, where the target receives scaled-down versions of updates applied to the *running* network with the intent to improve stability[31][67]. As a counter-reaction to this implied scale-up in network size and amount, recent support for the use of the less intensive alternative on-policy and/or decentralized methods such as decentralized PPO in MADRL can be identified[78][81][82].

Enhanced MADRL algorithms

In line with recommended use of recurrent networks, the work of Wang et al.[83] investigates the importance of recurrence by introducing it to actor networks, critic networks or both in fully- and partially-observable cooperative game setting. While their results indicate that recurrence leads to general improvement in game score for a partially observable setting, it is the introduction of a recurrent critic that leads to substantial improvement on the considered tasks. The authors attribute this to the apparent ability of recurrent critic to stabilize training by reducing the variance of rewards. Moreover, the recurrent critic seems to reduce the impact of the partial observable setting by dynamically/fluently combining information from different time steps, thereby acquiring an improved estimate of game dynamics. On the other hand, the need for sequenced data increases memory complexity and the associated recurrent networks are harder to train due to exploding/vanishing gradients over longer sequences[84].

To continue, MADDPG networks can be enhanced by integrating competitive game theoretic perspectives. Recall from Section 3.1, that in pursuit-evasion scenarios minimax optimization of the performance functions (Equation 3.2) minimizes loss against the worst adversarial policies and targets the Nash equilibrium. In their work, Li et al.[85] recognize that the original MADDPG formulation by Lowe et al.[77] might fail to minimize this perspective, resulting in brittle agents that are too sensitive to their adversaries' tactics. Therefore, they introduce the *minimax Multi-agent Deep Deterministic Policy Gradient* (M3DDPG) algorithm with the intent to robustify strategy determination in competitive scenarios.

In their work, M3DDPG is defined by explicit implementation of the minimax optimization objective into the update rule by assuming that all adversarial agents implement their own best policy. In addition, they reduce the computational effort required by approximating and optimizing a local linearization of the nonlinear Q-function. This latter step is called the *Multi-Agent Adversarial Learning* approach and inspired by Generative Adversarial Networks[86] for supervised learning tasks. From their results, they observe that agents trained by M3DDPG learned consistently more robust strategies than by MADDPG on a variety of tasks including a pursuit-evasion scenario, indicated by a retained higher strategy reward after continued training of adversaries.

Finally, a recent advancement in game theoretic MADRL is considered. In an approach by Perolat et al.[87], state-of-the-art performance is achieved applying DRL to the competitive sequential two-agent zero-sum imperfect information game of Stratego by explicitly targeting the Nash equilibrium in optimization through self-play. Specifically, their novel *DeepNash* system combines deep learning frameworks through residual neural networks with a game theoretical reinforcement algorithm named *Regularized Nash Dynamics* (R-NaD)[88]. The algorithm overcomes limitations of policy gradient methods in imperfect informa-

tion environments and achieves so without search, empirically supported for Stratego. The accomplishment of DeepNash on Stratego by Perolat et al.[87] further underlines the recent effectiveness of combining game theory and deep reinforcement learning approaches for multi-agent scenarios in observation-restricted environments.

R-NaD by Perolat et al.[88] defines an alternative three-stage approach applied over multiple iterations. After regularized redefinition of the current policy's reward, the updating approach guarantees learning convergence to a fixed point. In turn, this fixed point can be proven to converge to the Nash equilibrium if it is used as initiation for the next iteration, which is done until convergence. This example aligns with the aim of this chapter; to outline how combinations of deep- and reinforcement learning as well as game theory can provide superior strategies identified efficiently.

Alternative optimization methods

The main alternative approach to estimating the HJI equation and its solution can be found in *evolutionary learning* (EA). In this review, this methodology has been omitted thus-far due to reasons of scalability, efficiency and implementation complexity/practicality[89]. Evolutionary learning has been used to successfully identify robust policies in many types of scenarios[90], can overcome limitations of DRL (e.g. with regard to optimization guarantees)[89] or can even be combined with them into hybrid approaches[91][92]. However, its use in the context of differential pursuit-evasion games (with intricate dynamics) is unconventional.

To continue, the aforementioned reasons are briefly discussed. With regard to scalability and efficiency, EA can demand extensive computational effort exceeding that of DRL, in order to maintain a large population size containing a well-diversified set of agents that properly explores and avoids premature policy/behavior convergence. This potential problem of impractical scale is further emphasized for increasing complexities as invoked by an expanding state-action spaces as well as intricate dynamics such as partial observability, maneuverability constraints and/or adversarial settings. In addition, EA involves policy representation and mutation through a *genotype* encoding, which can become difficult to implement and track properly in practice as the network size and amount grows. Hence, due to the aforementioned reasons as well as the conventional use of DLR over EA methods in multi-agent pursuit-evasion differential games, the use of EA have not been further explored.

3.3. Insights from implementations

The previous section has outlined the prominent use of parameterized controllers in approximating the HJI solution through deep reinforcement learning. Therefore, the remainder of this section considers both approaches or hybrids and outlines key insights from relevant literature. Implementations for the pursuit-evasion context with implementation details are provided in Table 6.1 and Table 6.2 for single- and multi-agent setting respectively. Although not all of these implementations comprise a strict one-on-one pursuit-evasion scenario, they have been included to expand the set of available literature and provide a general overview of capabilities and deficiencies observed across approaches in this context.

In the case that an analytical solution to the pursuit-evasion game exists, its open-loop ($f(t)$) nature might not be practically feasible for use in controllers. This is due to the computational effort required to derive the solution at every cycle, as well as the inherent inability to properly stabilize the system compared to closed-loop systems ($f(x,t)$). An approach to overcome this impracticality is to optimize a parametric feedback controller such that it replicates optimal trajectories derived from the analytical solution[30][93][94][95], categorizable as a method of imitation learning[31].

The earlier work of Choi et al.[93] addresses this approach in the context of pursuit-evasion games and attains DNN controllers for both pursuer and evader trained on a dataset of optimal trajectories. In their work, deterioration in performance is observed for unseen agent states and non-optimal target trajectories. Although they augment the state space and trajectory types in an attempt to reduce these limitations, the reduction is clearly related to the augmentation size and the deterioration remains present. Notably, in another attempt to robustify their controllers, Choi et al. [93] combine their neural network controller guidance with the proportional navigation guidance law. The neural network guidance law is overruled by proportional navigation one whenever performance deterioration is observed due to unexpected target trajectories. In this setting, their hybrid controller is shown to improve over separate controllers and is capable of interception under a wider range of target trajectories, serving as an indication of the potential of hybrid/switched strategies in improving interception capabilities.

In this context, the work by Fu et al. [96] stands at the intersection between imitation and reinforce-

ment learning. In their work, a missile's pursuit strategy is learned using DDPG, where the replay buffer is initialized with expert experiences in the form of a collection of proportional navigation-based trajectories rather than self-explored ones. This approach is shown to improve training convergence speed; attributed to improved training progression during the initial, mainly exploratory, stages of the experiment. Similar to this approach, Li et al.[74] introduce an additional *self-imitation* optimization component to PPO meant to incite the exact replication of certain desired trajectories conducted in previous trials. Moreover, this work introduces an auxiliary target acceleration error minimization task to the main interception objective to explicitly encourage the extraction and subsequent incorporation of this information in the learned guidance law. Importantly, both works test the benefits of these augmentations through ablation studies and show how the augmentations improve interception efficacy and strategy robustness over imitated trajectories and/or conventional algorithm setups. All in all, the work by Li et al.[74] and Fu et al.[96] serve as examples of how conventional reinforcement learning algorithms can be augmented by auxiliary supervised learning setups or manipulation of the experience replay.

To continue, the literature provided in Table 6.1 and Table 6.2 serves as evidence for the potential of *adversarial*-, *curriculum*- and *meta-learning* for pursuit-evasion scenarios. Curriculum learning is shown to improve training stability and speed through the gradual and controlled increase in task complexity through a reduction in interception radius [97] and the range of possible dynamical system configurations for pursuer and evader [98] (e.g. relative speed). The latter is a form of meta-learning, which is shown to improve strategy robustness. It forms the main focus in the study by Gaudet et al.[73], where the learned strategy is shown to be adequately generalizable beyond the range of observed conditions during training. Moreover, the work by Wan et al.[99] augments DDPG with adversarial learning, which comprises an alternative method to improve strategy robustness. It is implemented by generating observation samples that purposefully invoke incorrect actions and subsequently focusing explicitly on the mitigation of the associated large losses during training.

Whereas single-agent DRL setups in Table 6.1 establish that DRL can improve over existing closed-loop pursuit control laws, the literature in Table 6.2 establishes that robustness can be further improved by considering a multi-agent setup for the pursuit-evasion scenario. In this case, robustness is defined with respect to the adversary's strategy. An example of this is the work by Cristino et al.[97] which shows that in multi-agent DRL setting the optimized pursuers are capable of capturing optimized and even human-operated evaders with greater efficacy than analytical guidance laws or a single-agent DRL setup. Similar conclusions on robustness against adversarial strategy configuration are drawn from comparisons in works by Gong et al.[98], Fu et al.[96], and Wan et al.[99]. Results from the work by Xiong et al.[100] are even more stringent; where they claim that for their experimental setup interception is guaranteed for an optimized pursuer with only a slight superiority in speed, even for an optimized evader with superior agility and utilizing intelligent maneuvering. Generally, the improved robustness aligns with the theory on differential games and the Nash equilibrium as discussed in Section 3.1, wherein this perspective implies optimization against the worst possible action of the adversary.

Besides the connection to game theory, the optimization progression in multi-agent reinforcement learning can be interpreted as a form of co-evolution from a biological perspective. Indeed, the work by Xiong et al.[100] addresses this; they observe that while existing pursuit control laws can only capture a non-optimized evader, an optimized pursuer does succeed in this setting. Hence, they state that not only has the pursuer evolved its strategy, but also the evader as indicated by its ability to escape from non-optimized pursuers. The observed co-evolution should be connected to the seminal work by Nolfi et al.[101] and their investigation into artificial evolutionary *arms-races*. Although this arms-race might produce complex behavior[102], it is not guaranteed to converge to the game optimal solution. Co-evolutionary setups potentially experience the *red-queen* effect[103] (i.e. premature stalled evolution/innovation) as well as catastrophic forgetting[31] leading to cyclical evolution (i.e. reversed innovation/declining complexity)[104].

3.4. Final remarks

This chapter has considered deep reinforcement learning as an approximation method for solutions to analytically intractable differential game formulations of pursuit and evasion scenarios. To conclude this chapter, the original sub-question is reconsidered. Through the differential game theory perspective, a game representation can be identified that encapsulates the dynamics between pursuer and evader agents. In addition, it provides insights into optimality conditions and states such

as the Nash equilibrium. In turn, this formulation can be combined with methods for multi-agent deep reinforcement learning (MADRL) which offer estimation procedures to approximate robust controller solutions. Throughout this chapter, the inherent complexity of pursuit-evasion tasks has become apparent. Therefore, general best practices for MADRL as well as example implementations have been summarized and identified, such as the careful consideration of agent's observability through the use of centralized critics as well as consideration of agent's optimal strategies in case asymmetric capabilities. Furthermore, in order to approximate game solutions adequately and in turn identify robust strategies, recurrent neural networks are consistently selected throughout relevant literature. Consequently, the subsequent chapter will consider their definition and recent advancements.

4 Neural pursuit controllers

In the previous chapter, the survey by Hernandez et al.[76] identifies the prominent use of recurrent networks in multi-agent games. Amongst recurrent neural networks, recent advancements have led to the formulation of a novel robust, scalable and interpretable type of recurrent network for control tasks[105]. These networks serve as an example of bio-inspired design and are known as *Liquid time-constant networks* and *neural circuit policies*. In light of this formulation, the remainder of this chapter addresses the following sub-questions with regard to these networks,

How can advancements in bio-inspired recurrent networks improve state-dependent controllers in terms of robustness, scalability and interpretability?

To properly address this question, this chapter initially contrasts recurrent network types to alternatives in network architectures, after which best practices and recent advancements are discussed, and relevant implementations are considered.

4.1. Artificial neural networks

The best learners around are biological neurons. Studies in neuroscience revealed that they have the following computational properties . The earliest practical imitation introduced the *perceptron* in the seminal work by McCulloh et al. [106] and simplifies neuron design through a weighted sum of inputs, after which a spiking synapse is introduced through a logic gate. Subsequent success on classification tasks served as demonstration of logical calculus with neuron replicating structures. Successive derivations have replaced the logic gate with the sigmoid transfer or *activation* function, interpretable as a static/constant (non-selective) synaptic feed-through mechanism. Consequently, such derivatives known as *artificial neural networks*(ANN) resemble biological neurons that do not spike, but allow for non-attenuated passive transmission in the analog domain [107].

Following the inception of the perceptron, singular units have been grouped into layers and connected to form networks, defining the multi-layer perceptron model. Numerous sequential layers of neurons can be constructed, with more than two hidden layers (i.e. neither input nor output layer) defined as a *deep neural network* (DNN) [64]. Although choice of specific architecture is task dependent, the prominence of these systems can be attributed to their theoretical universal function approximation property [108]. Importantly, one should recognize that these networks implement a naive interneuron connection protocol as well as responsibility indifferentiation. In other words, computational units are and remain arbitrarily and fully-connected to each other, even after model estimation. In addition, the capabilities/structure of single units is identical, rather than proposing altering neuron capabilities at different stages seen in biological networks.

While the transmissive properties of ANNs resemble non-spiking neurons, a clear difference in biophysical design is the exclusion of an internal state. This is explicitly defined in *recurrent neural networks* (RNNs) through the use of recurrent connections. Besides recovering biophysical neuron aspects, these recurrent connections establish a clear connection to feedback control design, as well as combating parameter gradient degradation (especially in deeper architectures) by circumventing the activation function[107]. The network resembles a The mathematical notation of both DNN and RNN are provided in Equation 4.1 and Equation 4.2 for the sigmoid activation function according to the notation by Radu et al.[107] and with connectivity visualized in Figure 4.1,

$$y_i^{t+1} = \sigma\left(\sum_{j=1}^{n} w_{ji}^t y_j^t + b_i^{t+1}\right), \quad \sigma(x) = \frac{1}{1+e^{-x}} \tag{4.1}$$

and RNNs,

$$x_i^{t+1} = -w_i x_i^t + \sum_{j=1}^{n} w_{ji}^t y_j^t, \quad y_j^t = \sigma\left(x_j^t + b_j^t\right) \tag{4.2}$$

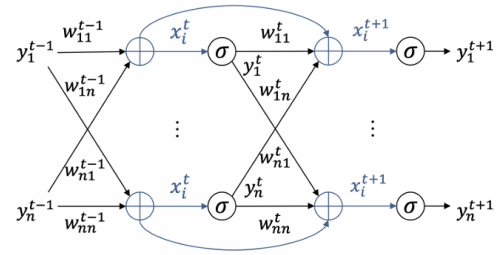


Figure 4.1: Deep and residual neural networks connection schematic. Image retrieved from [107].

where x and y define the pre- and post-synaptic state, respectively. The subscript ($_i$ or $_j$) denotes the neuron identifier, while the superscript (^{t}) is the layer identifier. w and b and define the variance and bias parameters, respectively.

Practical recurrent neural networks were first formulated as Hopfield networks by Hopfield [109] and made trainable based on concepts of back-propagation through time (BPTT) by Rumelhart et al. [110] with the concepts combined into vanilla RNN's by Elman [111].

Recurrent neural networks exhibit vanishing or exploding gradients which can benefit from memory gating as used in specialized discrete recurrent neural networks such as the *gated recurrent unit* (GRU) by Cho et al.[112] and the prominent *Long short-term memory* (LSTM) network by Hochreiter et al. [84]. In the recurrent networks, the recurrent connection (i.e. internal state; $-w_i x_i^t$ in RNNs) and specifically the weight parameters controls the strength of recurrence, leading to an implicit sense of memory depth governed by these parameters. Alternative network types define this memory depth explicitly by fixing the temporal horizon as observed in attention-based networks [113], temporal convolutions [114] and recently, retentive networks[115]. However, RNNs with their implicit time horizon are the choice when the temporal horizon is not known or varies, which is often the case in control tasks [116].

To continue, observe when $-w_i = 1$ holds in Equation 4.2 a parameterized forward Euler discretization of a continuous transformation (i.e. $\Delta x_{t+1} = f(\cdot)$) is described[117]. Consequently, the continuous transformation can be evaluated merely identified at discretize locations (i.e. time steps or layers), as visualized in on the left in Figure 4.2. In the limit of this discretization (i.e. $\Delta x_{t+\delta}$ as $\to 0$), the continuous transformation is parameterized directly. In such case, the continuous transformation can be evaluated at any point, as visualized in on the right in Figure 4.2. This type of model is known as a *neural ODE* (NODE) formulated by Chen et al.[117] and given according to the notation by Radu et al.[107] by,

$$\begin{aligned} \dot{x}_i(t) &= \sum_{j=1}^{n} w_{ji} y_{ji}(t) \\ y_{ji}(t) &= \sigma\left(x_j(t) + b_j\right). \end{aligned} \tag{4.3}$$

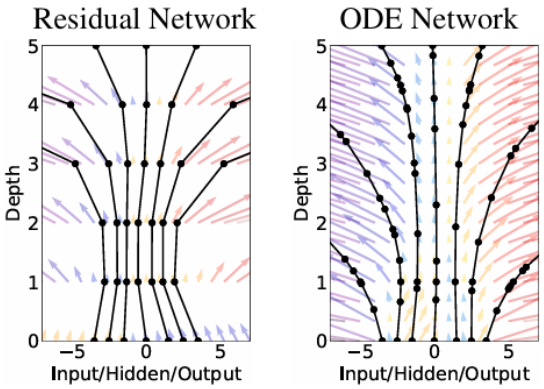


Figure 4.2: Possible evaluation points for discrete (left) and continuous (right) transformations as implied by residual/recurrent and ODE network types. *Residual neural network* (ResNet)[118] describes an recurrent connection equal to the identity. Image retrieved from [117].

In their original work, Chen et al.[117] identify parameter- and memory-efficiency as well as explicit modelling of continuous dynamics even at irregular time intervals as benefits over discretized alternatives, yet also recognize that these models are inherently harder to estimate with accuracy and stability governed by the ODE solver employed.

To continue, the *continuous-time recurrent network* (CT-RNN, Equation 4.4) can be formulated by augmenting the NODE (Equation 4.3) twice. First of all, introduction of a recurrent, leaking and stabilizing term, $-w_i x_i(t)$, interpretable as parameterized time-constant. Note that the implementation of NODE formulation is also possible for specialized discretized RNNs such as the LSTM[84] in turn leading to the definition of the (continuous-time) ODE-LSTM[119]. Secondly, by introducing term responsible for external input $u(t)$. All in all, the general case CT-RNN by Funahashi et al[120] can be represented according to the notation by Radu et al.[107] in Equation 4.4,

$$\begin{aligned}\dot{x}_i(t) &= -w_i x_i(t) + \sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t) \\ y_{ji}(t) &= w_{ji}\sigma\left(x_j(t) + b_{ji}^{x}\right) \\ z_{ji}(t) &= v_{ji}\sigma\left(u_j(t) + b_{ji}^{u}\right)\end{aligned} \tag{4.4}$$

where u defines the external system input and v as well as b^u the associated variance and bias parameters.

4.2. Liquid gated synapses

The general (non-autonomous) formulation of CT-RNN's are combined with these two fundamental design choices to acquire the *liquid time-constant neural networks* (LTC, Equation 4.5) by Hasani et al.[121], namely synaptic activation and linear gating. While a complete overview of intuition, methodology and proof as well as discussion for these augmentations for both CT-RNN and LTC is conducted by Radu et al. [107], this literature review summarizes the formulations' aspects. Ultimately, the general case LTC by Hasani et al.[121] can be represented according to the notation by Radu et al.[107] in Equation 4.5 respectively as,

$$\begin{aligned}C\dot{x}_i(t) &= w_{li}\left(e_{li} - x_i(t)\right) + \sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t) \\ y_{ji}(t) &= w_{ji}\sigma\left(a_{ji}^{x} x_j(t) + b_{ji}^{x}\right)\left(e_{ji} - x_i(t)\right) \\ z_{ji}(t) &= v_{ji}\sigma\left(a_{ji}^{u} u_j(t) + b_{ji}^{u}\right)\left(e_{ji} - x_i(t)\right)\end{aligned} \tag{4.5}$$

where a and e define two additional sets of variance and bias parameters respectively. Subscript ($_i$ or $_j$) denotes neuron identifier used to track connections in the network. Subscript l is used for terms related to the resting or *leaking* internal state of the neuron, addressed in the subsequent paragraph. Similar to the CT-RNN (Equation 4.4), this formulation has a stabilizing effect through the $-w_{li}x_i(t))$ term, yet is now re-centered by the e_{li} term in the difference $(e - x(t))$. Elaboration on this new difference term is discussed in the subsequent paragraphs as well.

The reason for is that the LTC formulation follows from biologically inspired electrical representation of a non-spiking neuron visualized in Figure 4.3. In this figure and Equation 4.5, Radu et al. [107] identify C as membrane conductance, w_{li} as leaking conductance and e_{li} and e_{ji} as resting and synaptic potential respectively. Hence, subscript l is used for terms related to the resting or *leaking* internal state of the neuron, related to non-directed outflow of the neuron (i.e. outflow not towards another neuron).

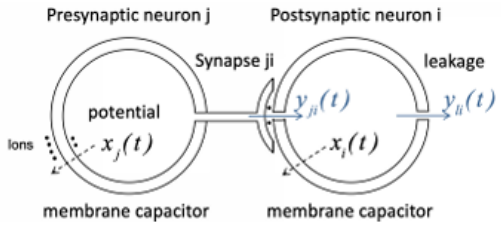


Figure 4.3: Electrical representation of non-spiking neuron. Note that this representation defines the neuron-synapse connection as autonomous, void of external inputs (u). Image retrieved from [107].

In the representation provided by Equation 4.5, the sigmoid transformed internal and external states itself should be interpreted as conductances which constitutes a current when multiplied with a difference in potential $(e - x(t))$. This inclusion of this term is known as *linear gating* and forms one of the fundamental design choices of LTC, besides synaptic activation discussed later. Notice that due to linear gating the information feed-through in the dynamical equation for $\dot{x}_i^t$ is now governed by the linear product of the synaptic nonlinear activation function, the *gate*, and this additional term representing the difference in potential.

Therefore, the inclusion of the difference term $(e - x(t))$ in the ultimate equation for y_j^t and z_j^t provides an additional direct feed-through mechanism for x_j^t) through the network. In fact, one might argue that the difference in potential $(e - x(t))$ forms the main path into the model, with the original synapse term $(\sigma(\cdot))$ reduced to (dynamic) scaling operation or *gating* of this difference term $(e - x(t))$. From a control perspective, this design choice definition is desirable traits as it represents an interpretable linear model with state dependent coefficients rather than a nonlinear transformation [107].

To continue, *Synaptic activation* is included by observing the inclusion of an additional pre-synaptic terms a in both equations for y and z. Furthermore, notice that these terms are specific to the synapse as represented by subscript ji. This aspect substantially increases number of parameters per neuron in the model and thereby the networks theoretical flexibility in terms of estimation capabilities [107]. Synaptic activation (SA) stands in contrast to the *neural activation* (NA), which defines non-unique parameter subset a within a neuron

(i.e. a_j^*) or its omission from Equation 4.5 entirely. Note that neural activation comprises the conventional approach to synapse parameterization and that all previous definitions (Equations 4.1-4.4) constitute the more parsimonious NA definition.

Besides comparison and discussion on models, the paper by Radu et al.[107] also investigates whether LTCs benefit from linear gating and synaptic activation. To this end they consider every possible combination of SA vs NA and linear vs no gating on a variety of time series prediction tasks. These task include straightforward prediction in sequential MNIST as well as more complex dynamics estimation in control tasks through the Half-Cheetah and Walker-2D environments. Their results indicate comparable performance for LTC to competitors on MNIST task and consistent superiority of LTCs to CT-RNN (i.e. non-gated LTC in this investgation) and other competitors on the control related tasks, regardless of activation configuration.

From this analysis the authors conclude that linear gating further improves accuracy over non-gated versions, besides the described benefits to interpretation and control. Generally, gating mechanisms have been identified as powerful tools in machine learning on variety of tasks to (dynamically) control the feed-through tolerances of specific information sources. With regard to time series, the LSTM[84] and GRU[112] utilize gating to offer dynamic control of memory depth. In another model designed for time series prediction, the attention[113]-based *temporal fusion transformer* by Lim et al.[122] the gating mechanism referred to as the *variable selection network* shows enhanced capabilities. Its gating mechanisms offers dynamic control of temporal depth, element- and/or group-wise throughput control of input features and permanent discarding of seemingly irrelevant features.

With regard to synaptic activation, Radu et al.[107] observe comparable performance of SA to NA when enforcing similar total parameter counts, yet SA achieves this performance with half the number of neurons. Hence, SA constitutes inherently smaller networks. In addition, the authors note that parameter amount of the LTC with synaptic activation is comparable to the competitive LSTM[84] (i.e. specialized RNN), yet emphasize that the LSTM's parameterization methodology is more complex.

4.3. Liquid time-constant networks

To continue, the liquid time-constant network (LTC) is redefined with its differential and recurrent equation according to the original definition by Hasani et al.[121], which is functionally identical to the previous representation (Equation 4.5). This formulation is given in Equation 4.6,

$$\begin{aligned}\frac{dx(t)}{dt} &= -\frac{x(t)}{\tau} + S(t) \\ &= -\left[\frac{1}{\tau} + f(x(t), I(t), t; \theta)\right] x(t) \\ &\quad + f(x(t), I(t), t; \theta) A. \\ \text{With } S(t) &= f(x(t), I(t), t; \theta)(A - x(t))\end{aligned} \tag{4.6}$$

where θ defines the complete set of parameters included in function f which comprises the synaptic combination of internal (x) and external inputs (I). A the resting synaptic reversal potential. This formulation can be used for taxonomic purposes, where *liquid* indicates the dynamic time-constant (i.e. $[\frac{1}{\tau} + f(x(t), I(t), t; \theta)]$) which adapts to the internal and external dynamics as described by parameterized function $f()$.

To continue, in the initial LTC definition by Hasani et al.[121] they theoretically prove the stable and bounded behaviour for this dynamic time-constant in this novel type of network as well as confirming the universal approximation property of functions for this type of network[123]. In fact, in the class of neural ordinary differential equation model representations, this type of network attains superior expressivity, defined as the inherent complexity scope as a function of parameter amount. This complexity scope of a model is measured by its ability to capture intricate patterns and dynamics with fewer parameters compared to alternative models. Although this is a rather abstract definition, the manner in which the authors assess complexity scope is visualized in Figure 4.4. This figure visualizes the changing arc-/trajectory-length for an circular input which is tracked to assess this scope.

The feat of improved expressivity is attributed to the continuous depth of these networks leading to parameter efficiency[117], as well as an increase in parameter amount through synaptic activation[107]. Specifically, the estimation capabilities of these networks as measured through increased computational depth exceeds that of competitors such as CT-RNN and LSTM by least a factor of 10^1 . Furthermore, the work by Hasani et al. presents results for time series prediction tasks similar to Radu et al. [107], yet for an expanded

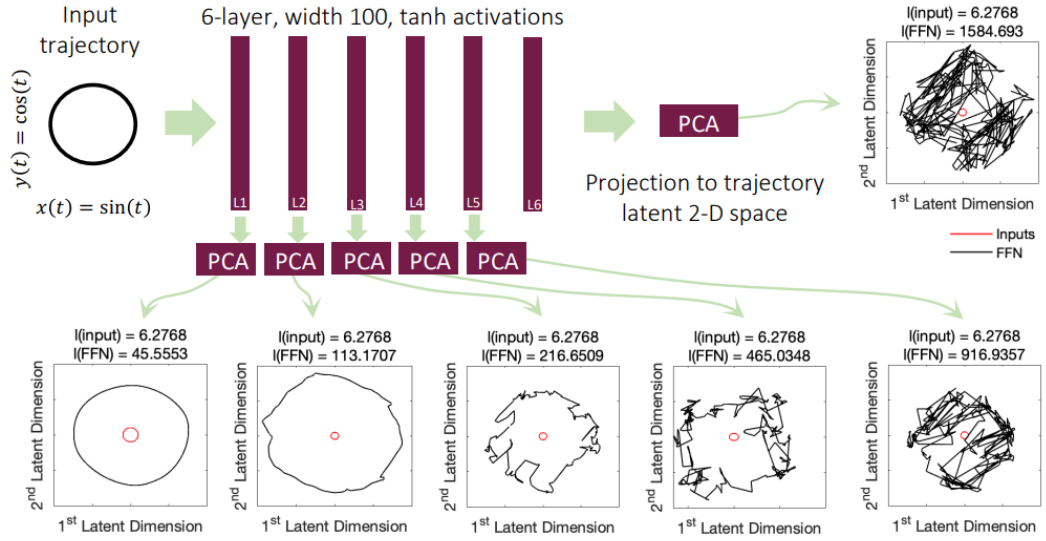


Figure 4.4: Visualization of complexity scope as measured through latent-spaced trajectory length (with dimension reduced through PCA). The initial trajectory is a circular motion, which is sequentially transformed into a more complex pattern through the layers. Notice how the latent trajectory becomes increasingly complex as it passes through the layers. Image retrieved from [121].

set of tasks. Their works align, highlighting the comparable or superior performance on benchmark datasets and strictly superior performance on control tasks of LTC compared to competitors.

Drawbacks of the LTC systems addressed by Hasani et al.[121] are threefold. First, the performance of LTCs is inherently connected to the ODE solver choice and implementation, with its limitations related to e.g. local/global accuracy. This is relevant for any ODE-based model (LTC, NODE, CT-RNN, etc.) and stands in contrast to (Euler) discretized representations. Secondly, the LTC imposes higher time and memory complexity compared to less sophisticated models such as NODE's. Finally, the LTCs present vanishing gradients, yet do not implement structures to limit this. For example, the discrete LSTM by Hochreiter[84] accounts for this through intricate memory gating structures. Hence, LTCs likely struggle at learning long-term (temporal) dependencies.

To complete this section, advancements to the liquid-time constant network class are considered. As stated before, the implementation of continuous networks is subject to ODE solver choice and their inherent limitations with regard to approximation errors, stability and computational efficiency. To circumvent inclusion of an ODE solver entirely, a closed-form solution approximation for liquid time-constant networks (Equation 4.6) is formulated by Hasani et al.[124] according to linear ODE theory as represented in Equation 4.7,

$$\begin{aligned} x(t) &= (x(0) - A)\mathrm{e}^{-\left[\frac{1}{\tau} + f(x(t), I(t), t; \theta)\right]t} \\ &\quad \cdot f(-x(t), -I(t), t; \theta) + A \\ x(t) &= B \odot \mathrm{e}^{-[w_\tau + f(x(t), I(t), t; \theta)]t} \\ &\quad \odot f(-x(t), -I(t), t; \theta) + A, \end{aligned} \tag{4.7}$$

where A defines the synaptic reversal potential, ultimately denoted by two terms (A&B) for increased flexibility.

Importantly, this closed-form definition does away with the inherent solver's local approximation error and its explicit time dependence reduces time complexity during estimation and inference of at least one order of magnitude without loss of accuracy[124]. They further prove this closed-form approximation is as expressive as its ODE-based version, thereby retaining the universal approximation property. This closed-form network definition in Equation 4.7 is further modified to reduce practical trainability issues such as vanishing gradients due to the recurrent nature and zero feed through during explosion in the exponential term. Hence, the authors ultimately introduce the *closed-form continuous depth* (CfC) network [124] through parameterized redefinition of terms A, B and the ex-

ponential function as in Equation 4.8,

$$x(t) = \underbrace{\sigma\left(-f\left(x(t), I(t), t; \theta_f\right)\mathbf{t}\right)}_{\text{time-continuous gating}} \odot g\left(x(t), I(t), t; \theta_g\right) + \underbrace{\left[1 - \sigma\left(-\left[f(x(t), I(t), t; \theta_f)\right]\mathbf{t}\right)\right]}_{\text{time-continuous gating}} \odot h\left(x(t), I(t), t; \theta_h\right). \tag{4.8}$$

where, compared to Equation 4.7, function g replaces the exponential term and B is replaced by gated $h(\cdot)$, introduced due to the relationship $B = x(0) - A$.

The CfC model is illustrated in Figure 4.5, with individual components discussed in the rest of this paragraph. These networks overcome said practical trainability issues through potential augmentation with memory architectures and time-continuous gating mechanisms represented through the sigmoid function ($\sigma()$) respectively. Furthermore, model flexibility is increased through redefinition of terms B and A into functions h and g separately, despite their relationship in Equation 4.7. The authors claim this separate definition allows for independent exploration of spatial and temporal features [124]. Finally, all parametric functions are subject to a connection to a prior *backbone* network allowing for shared feature extraction before branching out (see Figure 4.5).

To continue, the authors evaluate empirical results of the CfC against state-of-the-art discrete-time (e.g. LSTM) and continuous-time ODE-based (recurrent) networks (e.g. CT-RNN & NODE). Compared to these alternative models, various CfC definitions achieve consistent superiority for benchmark sequential data sets such as the high-dimensional human activity dataset and as well as control-related tasks such as modelling of physical dynamics of Walker2D in the MuJoCo physics engine.

Where the CfC overcomes drawbacks of the LTC related to the ODE-solver and time- & memory-complexity, it itself still exhibits vanishing gradients. To reduce this drawback for cases with clear long-term dependencies, the authors define an augmented network type named the *CfC-mixed memory RNN* (CfC-mmRNN) with memory gating similar to those used in the LSTM. Furthermore, the authors state verifying of proper neural flow and causality is likely easier for the ODE-based LTC than its closed-form approximation, the CfC. Furthermore, they reiterate that LTC and CfCs were fundamentally designed for causal time series tasks with unknown time horizons, often observed in control tasks. This stands in contrast to tasks related to natural language processing, where transformers leveraging self-attention[113] are conventionally used due to their powerful inference on known fixed temporal horizons.

4.4. Neural Circuit Policies

Throughout this chapter alternative definitions of the artificial neuron, the computational unit, have been discussed. In the work by Lechner et al.[105], another perspective to artificial neural redesign is considered through the *Neural Circuit Policy* (NCP) design algorithm, which describes an alternative approach to arbitrarily fully connecting neurons in a neural network. The design algorithm defines four layers of neurons and consists of four stages, as visualized in Figure 4.6. In their work, the neurons/computational units within these layers are exclusively chosen to be LTCs. As such, in the remainder of this review a NCP configuration indicates the wiring protocol combined with LTC units, unless otherwise defined.

The NCP algorithm defines a network wiring schematic, consisting of four hierarchical layers with varying amounts of inter connections as well as recurrent connections for command neurons. In between these layers, randomly initialized connections increase network sparsity compared to the fully connected alternative. This network wiring protocol follows from a preceding LTC-based network formulation named *ordinary neural circuits* (ONC) defined by Hasani et al.[125]. In this preceding work, a similar wiring protocol is formulated in an attempt at replicating the biological structure observed in the C. elegans nematode, reaching a parameter/weight sparsity of 77%.

It is important to recognize that the NCP/ONC wiring protocol comprises a constrained (Bernoulli-distributed) randomization between neurons in the network. Specifically, this constraint prohibits skip/reversing connections (e.g. sensory to command and vice versa) and excludes recurrent connections outside specific layers. In addition, sparsity is defined on a layer-specific basis leading to overall network sparsity rather than directly defining it on network basis uniformly across layers. For further details, one can refer to the NCP[105]/ONC[125] wiring algorithm.

To summarize, these protocols differ from unconstrained random connectivity. However, in their work defining the ONC by Hasani et al. [125], this biologically inspired wiring protocol is shown to achieve higher network max-flow rates than those achieved with less constrained and randomized connections. Furthermore, ONCs are

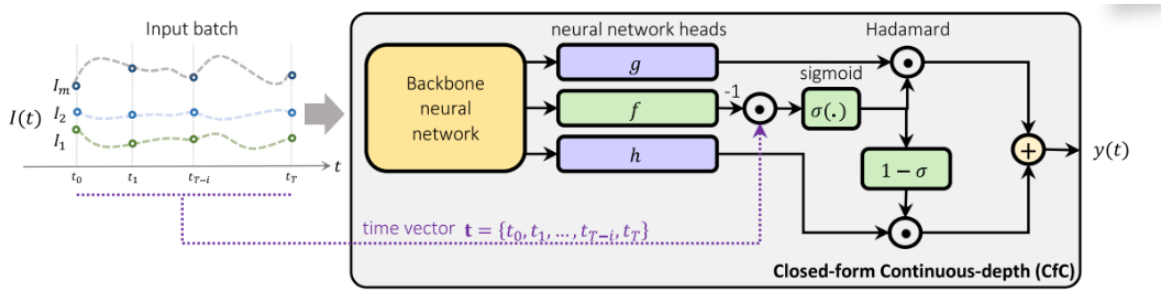


Figure 4.5: Closed-form continuous depth (CfC) network architecture according to the definition in Equation 4.8. Image retrieved from [124]

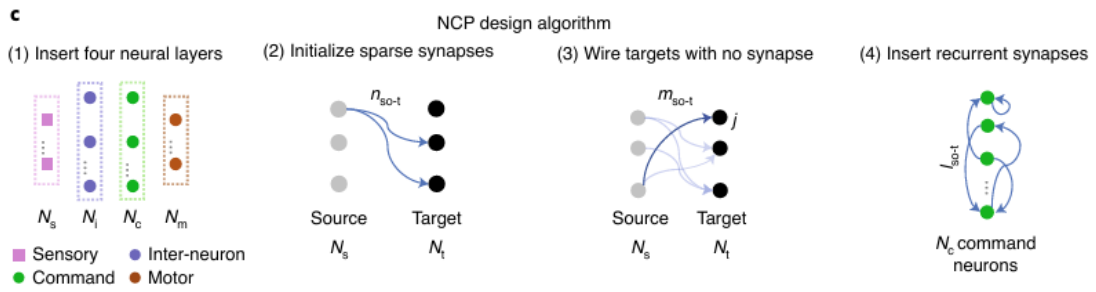


Figure 4.6: Visualized neural circuit policy design algorithm for undefined neuron type and layer size. Image retrieved from [105].

shown to consistently outperform networks with random connectivity as well as full connectivity empirically on benchmark reinforcement learning tasks such as Inverted Pendulum, Mountain car and Half-Cheetah. Consequently, the biological resemblance of the ONC formulation as well as the empirical superiority of these 'randomly' connected networks over their fully connected network counterparts deem it a *winning ticket* according to the *lottery ticket hypothesis*[126]. This theory defines the type of sparser network that is equal to or superior in performance to its dense counterpart with identical network architecture.

4.5. LTC-NCP implementations

Where previous sections described liquid networks and neural circuit policies, this section continues by considering all recent implementations. Relevant implementative works focus on analyzing *robustness*, *interpretability* and *scalability*.

Robustness and Causality

Besides the algorithm definition, the original work on NCPs by Lechner et al.[105] evaluates the performance of this wiring protocol combined with LTC's against a variety of alternative recurrent and ODE-based networks with the conventional fully-connected wirings/connections for the intricate control task of car lane keeping. Through imitation learning of recorded expert data in an offline setting, the trained controllers establish a mapping between visual inputs and output steering commands.

Results of this experiment in an online control setting show that task performance of the LTC with NCP wirings present a more interpretable attention profile and are superior in terms of out-of-distribution performance (generalizability) as well as robustness to various types of input perturbations. Importantly, this feat is achieved with an order of magnitude lower number of neurons, synapses and parameters to the closest scoring LSTM[84] and CT-RNN [120] system alternatives .

Furthermore, the original CfC paper definition by Hasani et al.[124] also considers this autonomous car lane-keeping task according to the same experiment methodology. In this work, the CfC model is compared to the ODE-based LTC (both with NCP wirings) as well as other benchmarking models. Remarkably, in this task the CfC is shown to perform comparably and be similarly robust to input noise perturbation, yet it achieves this performance at two magnitudes higher computational efficiency during estimation and inference than its ODE-based counterpart [124]. Finally, it is important to address that the CfC shows the lowest

parameter amount amongst the alternative model definitions (including LTC) and is therefore considered parameter efficient.

The generalizable and robust performance of CfC and LTC networks in combination with NCP is attributed to their inherent causal structure [121][124][105], as defined through the concept of directed acyclic graphs. The symbolic proof for this is threefold. First, LTC and CfC type of networks constitute a form of *dynamic causal models* (DCM); a type of network that captures the effect of both internal and external causes on dynamical systems[121][124][116]. Secondly, it is shown how the LTC/CfC networks type design give rise to causality through their forward- and backward pass as well as confirming the required ability to solve initial value problems with unique solutions[121][124][116]. Related to this, the work by Vorbach et al. [116] empirically supports theoretical results claiming that acquiring the causality characteristic is a feat, as general form discrete- and continuous-time recurrent models (as well as many derivatives thereof) are not causal models.

The empirical setting comprises a drone implementation in photo-realistic simulation of prominent recurrent and/or ODE-based algorithms contrasted against the NCP (with LTC) on a variety of visual-navigation tasks including static fly-to-target and more dynamic target following as well as (waypoint) hiking. The algorithms are trained through imitation learning of pure pursuit controller, where good and competitive performance is achieved in this passive open-loop case as measured through verification set loss. However, whenever tested in a closed-loop setting, NCPs show consistently and significantly superior performance with regard to task completion. This feat is observed through the algorithms' attention-profiles and attributed to the NCP's exclusive extraction of causal relationships resilient to external interventions (i.e. changes in environment conditions), where competitors' failure serves as evidence against their causal nature.

In another work, Chahine et al. [127] both LTC and CfC models with NCP wirings are utilized to train quadrotor controller for a vision-based fly-to-target task through imitation learning and evaluate it out-of-distribution without retraining (i.e. zero shot). This work can be considered as a continuation of the work by Vorbach et al.[116], where in this work a transfer from simulation to real environment is investigated.

The aim of this study is to evaluate performance robustness incited through severe distribution changes incited by drastic scenery changes resulting in completely new environments. Cases considered are changes in season, task conditions and background environment (e.g. urban vs. nature). Performance evaluation considers task success under for static and dynamic target tracking, stress testing through (further) perturbed inputs and further sensitivity analysis through increases in task range and invoking target rotations and occlusions. In fact, this work comprises a comparative study of all these metrics against a selection of recurrent and/or ODE-based networks.

The results of this work shows that the LTC and CfC type of networks attain an unparalleled level of performance robustness. To interpret and attribute this feat, the system's attention-profile (i.e. the system's focus) is analyzed which shows a more interpretable and improved ignorance of task irrelevant input features (e.g. background or noisy pixels).

Interpretability
In their original LTC-NCP experiments, Lechner et al.[105] raise network response interpretability by considering the controller's attention profile through analysis of principal components (PCA) of ultimate/intermediary layers, local score on performance metrics and visualized backward gradients revealing the controller's focus.

Recently, Wang et al. [128] further expanded this set of tools through an algorithm that extracts (logically interpretable) decision tree representations of the established controller mappings. In addition, they introduce a set of interpretability metrics that can measure disentanglement of trained controller dynamics (i.e. decoupled neurons). Subsequently, their methodology can be used to identify differentiated responsibilities of individual neurons and/or assess cross-neuron logic conflict/agreement.

The algorithm is successfully evaluated for a variety of control tasks, including trained controllers (including NCP) for car lane keeping for methodology similar to that of the original work on NCPs[105]. Finally, it is important to recognize that this work further establishes the impressive capabilities of the NCP-LTC network observed through consistent relatively high *neuron disentanglement* and decision accuracy as well as lower *cross-neuron conflict* across considered the control tasks compared to alternative architectures. Where in this setting, cross-neuron conflict defines the agreement between neurons on output commands (e.g. all agree to a speed increase) and neuron disentanglement implies specific neurons map to specific commands.

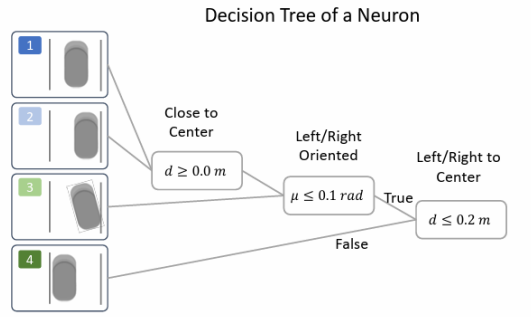


Figure 4.7: Interpretation of single neuron (of decoupled network) control mean response in decision tree format subject to certain local heading error (μ) and lateral deviation (d) states for NCP-LTC controller applied to lane keeping task. An interpretable decision tree follows from algorithm methodology by Wang et al. [128]. Image retrieved from [128].

Scalable and interpretable implementation
In a study by Tylkin et al. [129], NCPs are evaluated against MLP- and LSTM-based system architectures on two simulation-based tasks; the simpler *quadcopter dodgeball evasion* and more complex *fixed-wing canyon run*. Task complexity is further controlled by varying the amount of degrees of freedom. In these settings, agents are trained using PPO[71] in an end-to-end reinforcement learning setting, mapping information from visual inputs, depth sensors and vehicle state to commands.

While results of this study find comparable performance of NCPs in the drone dodgeball evasion task across varying degrees-of-freedom, NCPs are more performant on the canyon run task, specifically at more degrees-of-freedom. Furthermore, this study further establishes improved relative interpretability of NCPs at neuron-specific level decision-tree representations, similar in methodology to Wang et al. [128], as well as recording neuron activation under varying sensor conditions.

Finally, the study is completed by demonstrating onboard real-time inference as well as interpretability of NCP's trained on a dodgeball evasion task in a sim2real setting. To this end, the NCP system is implemented on a Raspberry Pi zero processor onboard a DJI Tello drone and neuron responses are recorded. In this setting, the drone presents generally consistent action policy alignment to simulation at both macro and neuron-specific level. Hence, this feat serves as evidence for the performant onboard implementation of NCP's while also retaining the level of command interpretability.

4.6. Final remarks

This chapter identifies advancements in recurrent models clearly and presents liquid networks with neural circuit policies as strong performers in control settings. To conclude this chapter, the original sub-question is reconsidered. Recent studies investigating implementations of these novel networks are few in number, yet further highlight their performance, robustness, interpretability and scalability. These results are connected to these networks' improved theoretical parameter efficiency/expressively as well as robustness to noise. In turn, these aspects are mainly attributed to their novel parameterization methodology (synaptic activation and liquid gating), the implementation of theory on neural differential equations and the model's inherent causal structure. In fact, besides the imitation of pure pursuit strategy, their use in interceptive control tasks is not well established. Therefore, they provide an interesting opportunity for future research in this context.

5 Conclusion

5.1. Challenges and Opportunities

The design of robust controllers capable of consistent interception present several challenges and opportunities for future research. In this section, we reconsider the foregone chapters from this perspective.

Most research in this review considers either controllers tested in simulation or controllers identified from recorded data, with little effort in validating the approach onboard real-life autonomous systems. In Chapter 2, it was hypothesized how the constant absolute target direction (CATD) or motion camouflage proportional guidance (MCPG) control law could theoretically encapsulate several pursuit strategies across varying pursuit conditions. In this case, an opportunity lies in the construction of unifying framework across pursuit strategies through a single control law. However, while the switching of strategies has been observed in predators, no consistent rules (e.g. heuristics) or dynamics have been established to explain when and why predators perform this switching behavior. Consequently, the challenge lies in the aspect that the design of a unifying controller cannot leverage inspiration from nature. In further regard to unification, it was discussed how the MCPG can be used for motion camouflage at finite focal point through the simulation study by Rano[21]. However, at this

point it is unknown whether this property is conditional on the evader trajectory and if the amount of angular slack (ϵ_m) required from the perspective of the evader is realistic. More fundamentally, a general limitation exists in that most research into natural behavior observed during pursuit and evasion scenarios is performed from the pursuer's perspective, rather than from both agents, which might define additional limiting conditions of the identified control laws.

Related to this is the investigation into the field of differential games of pursuit and interception discussed in Chapter 3 of this report. Together with deep reinforcement learning algorithms and novel artificial intelligence, this framework can be used to find solutions (i.e. controllers) for increasingly complex game formulations with aspects such as partial observability, nonlinear vehicle dynamics and constrained controllers. Therefore, an opportunity lies within the idea that these methodologies can provide new insights in natural behavior itself through analysis of the obtained solutions, such as the dynamics behind switching strategies. However, the challenge lies in the fact that optimal game states such as Nash equilibria need not exist, might be unreachable or cannot be evaluated to be optimal. Indeed, optimization routines replicating evolution such as multi-agent reinforcement learning might induce stalled or cyclically evolving agents. Consequently, this would imply no parallels of these obtained solutions to nature can be drawn.

Finally, this report considered novel recurrent network architectures in Chapter 4, named liquid neural networks and neural circuit policies. By improving parameter efficiency through bio-inspired solutions, these network types offer great potential in tackling well limitations of conventional neural network architectures, such as overfitting/noise susceptibility. In addition, their reduced network sizes and improved attention profile might improve interpretability. However, their use in multi-agent pursuit and evasion scenarios is not well established. The adversarial nature of this setting might require denser rather than sparser networks in order to provide robust control in all potential game states.

5.2. Conclusion

In this literature review, it was considered how insights from nature, game theory and robotics could be used to design controllers capable of robust pursuit and interception. To this end, in Chapter 2 interceptive pursuit strategies were evaluated, introducing the pure, deviated, and constant absolute target direction (CATD) strategies as well as formulating their associated control laws. Amongst these control laws, it was formulated how the CATD law could serve as inspiration in designing an effective onboard pursuit controller. This is due to its robustness to erratic evader motion as well as the minimization of perceivable visual cues, interception time and energy expenditure. Furthermore, it was discussed how CATD with state-dependent gain modulation can be used to dynamically switch between pursuit strategies in order to ensure interception.

In Chapter 3, it was examined how deep reinforcement learning could be used to identify controllers that approximate solutions to complex differential game formulations in pursuit-evasion scenarios. It highlighted the use of multi-agent deep reinforcement learning (MADRL) algorithms and recurrent neural networks (RNNs) to identify robust controllers, highlighting best practices and evaluating example implementations. Lastly, in Section 4.6 discussed recent advancements in recurrent models, particularly liquid networks with neural circuit policies, noting their performance, robustness, and scalability. These models, characterized by novel parameterization and theoretical foundations, potentially serve as more efficient and scalable alternatives to conventional RNN architectures, yet their performance in interceptive control tasks has not been well established.

5.3. Future Research

The challenges and opportunities throughout this report and summarized in Section 5.1 provide a research gap for future research into autonomous controller design for pursuit and evasion scenarios using deep reinforcement learning algorithms and novel artificial intelligence. Subsequently, these autonomous systems might be employed in practice onboard MAVs in order to contribute to sustainable agriculture through the combatting of insect pests. Besides these practical use cases, future research intends to shed light on the natural behavior of expert hunters through analysis of the identified controller dynamics. Therefore, in future research will focus on the objective;

to determine how bio-inspired artificial intelligence compare in interception efficacy of pursuit controllers to state-dependent gain modulation in the MCPG law for drone-based insect pest control.

Consequently, the corresponding research question is formulated as;

How do LTC-NCP networks compare to the

state-dependent gain modulated MCPG law in terms of interception efficacy onboard MAVs during pursuit of reactionary evaders for insect pest control?

This research question is addressed by handling the following sub-questions;

- What conditions warrant gain modulation or strategy switches during pursuit of reactionary evaders?
- How do LTC-NCP based controllers compare to the state-dependent gain modulated MCPG law or alternative network parameterization schemes?
- How do simultaneous and adversarial optimization of pursuit and evader objectives through multi-agent reinforcement learning compare to evolutionary principles?
- What evasive strategies bring forth limiting conditions in predatory pursuit?

In the formulation of an answer to these sub-questions, subsequent research will conclude on the feasibility of utilizing multi-agent reinforcement learning and novel bio-inspired artificial intelligence to optimize controllers in pursuit and evasion scenarios. It will attempt to provide design principles for effective pursuit controller design for autonomous control onboard MAVs and can potentially outline insights into natural predator behavior.

6 Literature review tables

Source	Setup	Input	Output	Con-troller	I/O Conditions	Estimation/op-timization	Objective/reward	Adversary dynamics
[73]	3D-P1-e1	Line of sight angles and their rates	Thrust	RNN	ideal inputs & outputs	PPO under varying scenario conditions (meta RL)	rotation and rotational rate error as well as interception	Evader capable of performing bang-bang or barrel roll maneuver, besides evasion control law for acceleration.
[130]	2D-P1-e1 (MDT)	Line of sight angle (rate), range, and closing velocity	Acceleration	RNN	Input noise	DDPG variant	rotation and rotational rate error as well as interception and minimal error consumption	The adversary executes a square-wave maneuver or a PNG law attempting to reach a target position.
[131]	3D-P1-e1	Line-of-sight angles, line-of-sight angular velocities, angle of attack, and sideslip angle, (change of) distance.	Pitch and yaw fin-deflection angles	DNN	Inputs ideal, outputs subject to delayed first order TF.	DDPG	Besides interception, rewards for zero rotational rate of LoS , miss-distance, control deflection and a deep dual filter to improve convergence speed	Evader follows sinusoidal motion in the y-direction
[75]	2D/3D-p1-E1	Line-of-sight (LoS) angles and their rates of change	Thrust in 2 axes.	DNN	No noise or delay is considered	PPO	Reward focuses on evasion success and control effort.	Pursuer ZEM law
[132]	2D-pn-E1	Consecutive images of absolute grid positions	Discretized direction.	CNN	Ideal inputs & outputs. Discretized grid with particle environment	DQN	Rewards focus on distance and interception.	Adversary pursuer team follows predefined strategies.
[96]	2D-P1-e1	Own attitude and velocity as well as relative information and distance.	Acceleration and rotational rates	DNN	No noise or delay is considered. Rotational rates and velocities subject to limits, subsequently implies nonholomic movement/turn radius.	DDPG with imitation learning	Reward focusses distance and capture success.	Uniform linear motion or reverse pure/deviated pursuit escape strategy implemented through PN with gain =1 (p3. 5). Evader maintains constant velocity.
[133]	2D-P1-e1	Roll of itself and of adversary. Relative position and relative yaw. Notice no z-axis related content.	Discretized Roll and velocity commands	RNN	Inputs reflect true state; no explicit mention of noise/delay/ Discretized outputs have limits.	A3C	Reward for getting adversary in the field of view (dog-fight setup)	Opponent modeled with Greedy Shooter (GS) behavior and constant velocity;
[74]	3D-P1-e1	Relative distance and its rate as well as angular errors (pitch & yaw) and their rate.	Acceleration commands In 2 axes	DNN	Inputs include noise/delay, also every episode certain sensors are masked. Outputs ideal.	augmented PPO with auxilarly learning, self imitation and exploration regulation.	Reward based on distance, interception angle	Non-learnt evader strategy. Target initially moves allong straight trajectory with constant velocity. After pursuer detection, the target choses a random direction and escapes with fixed acceleration.

Table 6.1: Pursuit-evasion implementations with a single optimized agent. The *setup* contains a code describing the number of dimensions (D), as well as the number of pursuer (P/p) and evader (E/e) agents, where N indicates multiple > 1. A capital letter (e.g. E), as opposed to a lower case one (e.g. e), indicates a data-driving technique is used to identify the controller for the respective agent. Additional abbreviations for models and algorithms have been introduced in Chapter 3 and Chapter 4. The order of sources in these tables reflects that of the order of discussion in these chapters.

Source	Setup	Input	Output	Controller	I/O Conditions	Estimation/optimization	Objective/reward	Adversary dynamics
[98]	2D-P1-E1 (MDT)	distance, closing speed, line-of-sight angle & rate	Effective navigation gains for a predefined proportional guidance law	DNN	Kalman filtered inputs, ideal outputs.	PPO algorithm using curriculum learning	Interception radius and distance	Intelligent adversary
[97]	2D-Pn-e1	(Change of) distance and heading angle (rate) of the evader. In addition, relative position and heading of neighboring pursuers	Rotational rate	DNN	No noise or delay in inputs. Pursuer non-holomic, evader holomic.	Variant of MADDPG with curriculum learning	Formation rewards and capture success	Intelligent adversary
[99]	2D-Pn-E1	own and relative position, attitude and velocity.	Accelerations in 2 axes.	DNN	Noisy inputs, ideal outputs. Obstacles in arena.	MADDPG with adverserial learning.	Reward focusses on distance and interception success, as well as collision penalty.	Intelligent more maneuverable evader
[134]	2D-Pn-E1	Own and other's discretized position.	Discretized direction.	DNN	Ideal inputs & outputs. Discretized grid with particle environment.	MADDPG and (game theoretic inspired) QMIX	Reward focusses on distance, success and collision.	Intelligent evader with higher speed.
[100]	2D-P1-E1	Own and other's position and velocity.	Acceleration in 2 axes	DNN	Ideal inputs and outputs.	MADDPG	Rewards focus on distance and interception.	Intelligent adversary or adversary using proportional navigation.
[135]	3D-P1-E1	Relative position	Desired attitude	Fuzzy Policies	Noisy inputs. Outputs modelled through non-ideal dynamics systems.	Q-learning with temporal difference error	distance and closing speed	Intelligent adversary with similar control system

Table 6.2: Pursuit-evasion implementations with multiple optimized agents. The *setup* contains a code describing the number of dimensions (D), as well as the number of pursuer (P/p) and evader (E/e) agents. A capital letter (e.g. E), as opposed to a lower case one (e.g. e), indicates a data-driving technique is used to identify the controller for the respective agent. Additional abbreviations for models and algorithms have been introduced in Chapter 3 and Chapter 4. The order of sources in these tables reflects that of the order of discussion in these chapters.

Part IV

Closure

The intent of this research is to contribute to a more sustainable form of agriculture with reduced needs for insecticides through the development and assessment of an optimization framework to identify robust controllers for autonomous MAVs that can consistently intercept insect pests. To this end, this report has provided a comprehensive review of literature focussing on natural predator behavior, optimization in game theory and advancements in neural nonlinear controllers. Subsequently, this research implemented the aforementioned framework and has demonstrated the successful optimization of adversarial strategies between a drone pursuer and an insect-inspired evader using multi-agent deep reinforcement learning.

This report is finalized by concluding on the original research questions and providing an overview of the study's limitations and potential directions for future research. The research questions posed in Part 1 are repeated below for convenience.

Research Question 1

Do parameterized nonlinear controllers for quadcopter pursuers identified through multi-agent deep reinforcement learning outperform the single-agent alternative in terms of the interception rate of insect-like evaders?

and,

Research Question 2

Does natural predator behavior emerge from the co-evolution between a quadcopter pursuer and insect-like evader in simulated games of pursuit and evasion?

To address the first research question, the results in this report show that the drone pursuer is consistently able to pursue and intercept a reactive insect-inspired evader as well as recordings of actual insect targets, achieving interception rates of 55% and 94% on these respective tasks. In comparison, pursuers alternatively optimized against non-reactive evaders or reactive drone-like evaders with symmetric capabilities, achieve an interception rate of only 42% for the same insect target recordings. Despite these promising results, it is concluded that further research is needed to formally establish the superiority of multi-agent optimization in this asymmetric game scenario.

With regard to the second research question, it was identified how natural pursuit behaviors, such as motion camouflage and pure pursuit, indeed emerged after the co-evolution between the drone pursuer and insect-like evader in simulation. Specifically, the results in this report outline that the drone pursuer mainly implements pure-pursuit as well as motion camouflage to some degree. Consequently, comparisons can be drawn to the hunting strategy of dragonfly as well as other killer flies. These need to be further investigated.

Several limitations need to be addressed. While the study attempts to carefully match the capabilities of the pursuer and evader agents to their realistic counterparts, it has also assumed an unlimited field-of-view for both agents. This choice is biologically unrealistic, as natural hunters are constrained by visual tracking limitations, which is expected to have an impact on the emergent behavior in the simulation. Additionally,

the study has not fully explored the effects of and sensitivity to more challenging factors such as observation noise, motor dynamics, or controller implementation capabilities. Although the evader model provides a qualitative match to real-world insect dynamics, its accuracy can be improved through a thorough exercise in system identification. Moreover, the study did not confirm whether the controllers reached a local stable or a global equilibrium game state, such as a Nash equilibrium. Finally, expanding the research to consider end-to-end controllers without low-level incorporation could help overcome limitations of current control incorporation and could emphasize the strengths of the proposed solution.

Future work should focus on addressing these limitations. Importantly, practical testing is necessary to assess the real-world feasibility of the proposed strategies in reducing the use of insecticides in greenhouses.

References

[1] PATS Indoor Drone Solutions. “Monitor and eliminate pests in your greenhouse crops”. In: *PATS Website* (2024). URL: `https://www.pats-drones.com`.

[2] Richard J Bomphrey et al. “Flight of the dragonflies and damselflies”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 371.1704 (2016), p. 20150389.

[3] Norbert Boeddeker et al. “Chasing a dummy target: smooth pursuit and velocity control in male blowflies”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1513 (2003), pp. 393–399.

[4] MF Land. “Chasing and pursuit in the dolichopodid fly Poecilobothrus nobilitatus”. In: *Journal of Comparative Physiology A* 173 (1993), pp. 605–613.

[5] S Pal. “Dynamics of aerial target pursuit”. In: *The European Physical Journal Special Topics* 224.17-18 (2015), pp. 3295–3309.

[6] Michael F Land et al. “Chasing behaviour of houseflies (Fannia canicularis) A description and analysis”. In: *Journal of comparative physiology* 89 (1974), pp. 331–357.

[7] SW Zhang et al. “Visual tracking of moving targets by freely flying honeybees”. In: *Visual neuroscience* 4.4 (1990), pp. 379–386.

[8] Paloma T Gonzalez-Bellido et al. “Target detection in insects: optical, neural and behavioral optimizations”. In: *Current opinion in neurobiology* 41 (2016), pp. 122–128.

[9] Suzanne Amador Kane et al. “Falcons pursue prey using visual motion cues: new perspectives from animal-borne cameras”. In: *Journal of Experimental Biology* 217.2 (2014), pp. 225–234.

[10] Akiko Mizutani et al. “Motion camouflage in dragonflies”. In: *Nature* 423.6940 (2003), pp. 604–604.

[11] Kaushik Ghose et al. “Echolocating bats use a nearly time-optimal strategy to intercept prey”. In: *PLoS biology* 4.5 (2006), e108.

[12] Trevor J Wardill et al. “A novel interception strategy in a miniature robber fly with extreme visual acuity”. In: *Current Biology* 27.6 (2017), pp. 854–859.

[13] Ermin Wei et al. “Pursuit and an evolutionary game”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 465.2105 (2009), pp. 1539–1559.

[14] Eric W Justh et al. “Steering laws for motion camouflage”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 462.2076 (2006), pp. 3629–3643.

[15] PV Reddy et al. “Motion camouflage in three dimensions”. In: *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE. 2006, pp. 3327–3332.

[16] Eric W Justh et al. “Natural frames and interacting particles in three dimensions”. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 2841–2846.

[17] Samuel T Fabian et al. “Interception by two predatory fly species is explained by a proportional navigation feedback controller”. In: *Journal of The Royal Society Interface* 15.147 (2018), p. 20180466.

[18] Kevin S Galloway et al. “Motion camouflage in a stochastic setting”. In: *2007 46th IEEE conference on decision and control*. IEEE. 2007, pp. 1652–1659.

[19] PV Reddy et al. “Motion camouflage with sensorimotor delay”. In: *2007 46th IEEE conference on decision and control*. IEEE. 2007, pp. 1660–1665.

[20] Vidya Raju et al. “Motion camouflage in the presence of sensory noise and delay”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 2846–2852.

[21] Iñaki Rañó. “On motion camouflage as proportional navigation”. In: *Biological cybernetics* 116.1 (2022), pp. 69–79.

[22] Puduru Viswanadha Reddy. *Steering laws for pursuit*. University of Maryland, College Park, 2007.

[23] Reuben Strydom et al. “Biologically inspired interception: A comparison of pursuit and constant bearing strategies in the presence of sensorimotor delay”. In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2015, pp. 2442–2448.

[24] Qiancheng Zhao et al. “Performance analysis of motion camouflage guidance law against maneuvering target”. In: *IEEE Transactions on Aerospace and Electronic Systems* (2023).

[25] Andrew James Anderson et al. “Humans deceived by predatory stealth strategy camouflaging motion”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.suppl_1 (2003), S18–S20.

[26] Paul Glendinning. “The mathematics of motion camouflage”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 271.1538 (2004), pp. 477–481.

[27] Neryahu A Shneydor. *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998.

[28] NE Carey et al. “Energy-efficient motion camouflage in three dimensions”. In: *arXiv preprint arXiv:0806.1785* (2008).

[29] Mandyam V Srinivasan et al. “Strategies for active camouflage of motion”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 259.1354 (1995), pp. 19–25.

[30] Andrew James Anderson et al. “Model of a predatory stealth behaviour camouflaging motion”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1514 (2003), pp. 489–495.

[31] Richard S Sutton et al. *Reinforcement learning: An introduction*. MIT press, 2018.

[32] Mandyam V Srinivasan. “Where paths meet and cross: navigation by path integration in the desert ant and the honeybee”. In: *Journal of Comparative Physiology A* 201 (2015), pp. 533–546.

[33] Iñaki Rañó. “An optimal control strategy for two-dimensional motion camouflage with non-holonimic constraints”. In: *Biological cybernetics* 106 (2012), pp. 261–270.

[34] Steven D Wiederman et al. “A predictive focus of gain modulation encodes target trajectories in insect vision”. In: *Elife* 6 (2017), e26478.

[35] Claire Plunkett et al. “Modeling Coordinate Transformations in the Dragonfly Nervous System”. In: *Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference*. 2023, pp. 6–10.

[36] Frances S Chance. “Interception from a Dragonfly Neural Network Model”. In: *International Conference on Neuromorphic Systems 2020*. 2020, pp. 1–5.

[37] Matteo Mischiati et al. “Internal models direct dragonfly interception steering”. In: *Nature* 517.7534 (2015), pp. 333–338.

[38] Leandre Varennes et al. “Two pursuit strategies for a single sensorimotor control task in blowfly”. In: *Scientific reports* 10.1 (2020), p. 20762.

[39] Ashley N Peterson et al. “Pursuit and evasion strategies in the predator–prey interactions of fishes”. In: *Integrative and comparative biology* 61.2 (2021), pp. 668–680.

[40] Brett R Fajen et al. “Behavioral dynamics of intercepting a moving target”. In: *Experimental Brain Research* 180.2 (2007), pp. 303–319.

[41] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[42] Isaac E Weintraub et al. “An introduction to pursuit-evasion differential games”. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 1049–1066.

[43] João P Hespanha. *Noncooperative game theory: An introduction for engineers and computer scientists*. Princeton University Press, 2017.

[44] Russ Tedrake. *Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023. URL: `https://underactuated.csail.mit.edu`.

[45] Lewis Meier. “A new technique for solving pursuit-evasion differential games”. In: *IEEE Transactions on Automatic Control* 14.4 (1969), pp. 352–359.

[46] Ritwik Bera et al. “A comprehensive differential game theoretic solution to a game of two cars”. In: *Journal of Optimization Theory and Applications* 174 (2017), pp. 818–836.

[47] I Greenfeld. “A differential game of surveillance evasion of two identical cars”. In: *Journal of optimization theory and applications* 52 (1987), pp. 53–79.

[48] Joseph Lewin et al. “The surveillance-evasion game of degree”. In: *Journal of Optimization Theory and Applications* 16 (1975), pp. 339–353.

[49] Jaime F Fisac et al. “The pursuit-evasion-defense differential game in dynamic constrained environments”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 4549–4556.

[50] Dave Wilson Oyler. “Contributions To Pursuit-Evasion Game Theory.” PhD thesis. 2016.

[51] Kalyanam Krishnamoorthy et al. “Pursuit on a graph using partial information”. In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 4269–4275.

[52] Meir Pachter et al. “A stochastic homicidal chauffeur pursuit-evasion differential game”. In: *Journal of Optimization Theory and Applications* 34 (1981), pp. 405–424.

[53] Simone Battistini et al. “Differential games missile guidance with bearings-only measurements”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.4 (2014), pp. 2906–2915.

[54] Othusitse Basimanebotlhe et al. “Stochastic optimal control to a nonlinear differential game”. In: *Advances in Difference Equations* 2014.1 (2014), pp. 1–14.

[55] FL Chernousko et al. “Some differential games with incomplete information”. In: *IFIP Technical Conference on Optimization Techniques*. Springer. 1974, pp. 445–450.

[56] Yakoov Yavin. “A pursuit-evasion differential game with noisy measurements of the evader’s bearing from the pursuer”. In: *Journal of optimization theory and applications* 51 (1986), pp. 161–177.

[57] G Hexner. “A differential game of incomplete information”. In: *Journal of Optimization Theory and Applications* 28 (1979), pp. 213–232.

[58] Nigel Greenwood. “A differential game in three dimensions: The aerial dogfight scenario”. In: *Dynamics and Control* 2.2 (1992), pp. 161–200.

[59] J Shinar et al. “Recent advances in optimal pursuit and evasion”. In: *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. IEEE. 1979, pp. 960–965.

[60] Josef Shinar. “Solution techniques for realistic pursuit-evasion games”. In: *Advances in control and dynamic systems* 17 (1981), pp. 63–124.

[61] Fumiaki Imado et al. “A method to solve missile-aircraft pursuit-evasion differential games”. In: *IFAC Proceedings Volumes* 38.1 (2005), pp. 176–181.

[62] Josef Shinar et al. “A pursuit-evasion game with hybrid pursuer dynamics”. In: *European Journal of Control* 15.6 (2009), pp. 665–684.

[63] Josef Shinar et al. “A pursuit-evasion game with hybrid evader dynamics”. In: *2009 European Control Conference (ECC)*. IEEE. 2009, pp. 121–126.

[64] Ian Goodfellow et al. *Deep learning*. MIT press, 2016.

[65] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *nature* 518.7540 (2015), pp. 529–533.

[66] Christopher JCH Watkins et al. “Q-learning”. In: *Machine learning* 8 (1992), pp. 279–292.

[67] David Silver et al. “Deterministic policy gradient algorithms”. In: *International conference on machine learning*. Pmlr. 2014, pp. 387–395.

[68] Volodymyr Mnih et al. “Asynchronous methods for deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[69] Peter Jin et al. “Regret minimization for partially observable deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2018, pp. 2342–2351.

[70] Shengyi Huang et al. “A2C is a special case of PPO”. In: *arXiv preprint arXiv:2205.09123* (2022).

[71] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[72] Nicolas Heess et al. “Emergence of locomotion behaviours in rich environments”. In: *arXiv preprint arXiv:1707.02286* (2017).

[73] Brian Gaudet et al. “Reinforcement learning for angle-only intercept guidance of maneuvering targets”. In: *Aerospace Science and Technology* 99 (2020), p. 105746.

[74] Weifan Li et al. “Missile guidance with assisted deep reinforcement learning for head-on interception of maneuvering target”. In: *Complex & Intelligent Systems* 8.2 (2022), pp. 1205–1216.

[75] Mengda Yan et al. “Ballistic Missile Midcourse Intelligent Maneuver Strategy Based on PPO Algorithm”. In: *International Conference on Guidance, Navigation and Control*. Springer. 2022, pp. 5169–5178.

[76] Pablo Hernandez-Leal et al. “A survey and critique of multiagent deep reinforcement learning”. In: *Autonomous Agents and Multi-Agent Systems* 33.6 (2019), pp. 750–797.

[77] Ryan Lowe et al. “Multi-agent actor-critic for mixed cooperative-competitive environments”. In: *Advances in neural information processing systems* 30 (2017).

[78] Xueguang Lyu et al. “Contrasting centralized and decentralized critics in multi-agent reinforcement learning”. In: *arXiv preprint arXiv:2102.04402* (2021).

[79] Jianhao Wang et al. “Qplex: Duplex dueling multi-agent q-learning”. In: *arXiv preprint arXiv:2008.01062* (2020).

[80] Stefano V Albrecht et al. “Multi-agent reinforcement learning: Foundations and modern approaches”. In: *Massachusetts Institute of Technology: Cambridge, MA, USA* (2023).

[81] Chao Yu et al. “The surprising effectiveness of ppo in cooperative multi-agent games”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24611–24624.

[82] Jakub Grudzien Kuba et al. “Settling the variance of multi-agent policy gradients”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 13458–13470.

[83] Rose E Wang et al. “R-MADDPG for partially observable environments and limited communication”. In: *arXiv preprint arXiv:2002.06684* (2020).

[84] Sepp Hochreiter et al. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[85] Shihui Li et al. “Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 4213–4220.

[86] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[87] Julien Perolat et al. “Mastering the game of Stratego with model-free multiagent reinforcement learning”. In: *Science* 378.6623 (2022), pp. 990–996.

[88] Julien Perolat et al. “From Poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8525–8535.

[89] Amjad Yousef Majid et al. “Deep reinforcement learning versus evolution strategies: A comparative survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[90] Akbar Telikani et al. “Evolutionary machine learning: A survey”. In: *ACM Computing Surveys (CSUR)* 54.8 (2021), pp. 1–35.

[91] Hui Bai et al. “Evolutionary reinforcement learning: A survey”. In: *Intelligent Computing* 2 (2023), p. 0025.

[92] Pengyi Li et al. “Bridging Evolutionary Algorithms and Reinforcement Learning: A Comprehensive Survey”. In: *arXiv preprint arXiv:2401.11963* (2024).

[93] Han-Lim Choi et al. “Neural network guidance based on pursuit-evasion games with enhanced performance”. In: *IFAC Proceedings Volumes* 35.1 (2002), pp. 103–108.

[94] Mikhail Khachumov et al. “Modeling the Solution of the Pursuit–Evasion Problem Based on the Intelligent–Geometric Control Theory”. In: *Mathematics* 11.23 (2023), p. 4869.

[95] Robin Ferede et al. “End-to-end neural network based optimal quadcopter control”. In: *Robotics and Autonomous Systems* 172 (2024), p. 104588.

[96] Xiaowei Fu et al. “A UAV pursuit-evasion strategy based on DDPG and imitation learning”. In: *International Journal of Aerospace Engineering* 2022 (2022), pp. 1–14.

[97] Cristino De Souza et al. “Decentralized multi-agent pursuit using deep reinforcement learning”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4552–4559.

[98] Xiaopeng Gong et al. “Intelligent game strategies in target-missile-defender engagement using curriculum-based deep reinforcement learning”. In: *Aerospace* 10.2 (2023), p. 133.

[99] Kaifang Wan et al. “An improved approach towards multi-agent pursuit–evasion game decision-making using deep reinforcement learning”. In: *Entropy* 23.11 (2021), p. 1433.

[100] Hao Xiong et al. “A Dynamics Perspective of Pursuit-Evasion Games of Intelligent Agents with the Ability to Learn”. In: *2022 41st Chinese Control Conference (CCC)*. IEEE. 2022, pp. 7082–7087.

[101] Stefano Nolfi et al. “Coevolving predator and prey robots: Do “arms races” arise in artificial evolution?” In: *Artificial life* 4.4 (1998), pp. 311–335.

[102] Trapit Bansal et al. “Emergent complexity via multi-agent competition”. In: *arXiv preprint arXiv:1710.03748* (2017).

[103] Leigh Van Valen. “The red queen”. In: *The American Naturalist* 111.980 (1977), pp. 809–810.

[104] Stefano Nolfi. “Co-evolving predator and prey robots”. In: *Adaptive Behavior* 20.1 (2012), pp. 10–15.

[105] Mathias Lechner et al. “Neural circuit policies enabling auditable autonomy”. In: *Nature Machine Intelligence* 2.10 (2020), pp. 642–652.

[106] Warren S McCulloch et al. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[107] Julian Lemmel et al. “On the Benefits of Biophysical Synapses”. In: *arXiv preprint arXiv:2303.04944* (2023).

[108] Kurt Hornik et al. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

[109] John J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[110] David E Rumelhart et al. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[111] Jeffrey L Elman. “Finding structure in time”. In: *Cognitive science* 14.2 (1990), pp. 179–211.

[112] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[113] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[114] Shaojie Bai et al. “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”. In: *arXiv preprint arXiv:1803.01271* (2018).

[115] Yutao Sun et al. “Retentive network: A successor to transformer for large language models”. In: *arXiv preprint arXiv:2307.08621* (2023).

[116] Charles Vorbach et al. “Causal navigation by continuous-time neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12425–12440.

[117] Ricky TQ Chen et al. “Neural ordinary differential equations”. In: *Advances in neural information processing systems* 31 (2018).

[118] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[119] Mathias Lechner et al. “Learning long-term dependencies in irregularly-sampled time series”. In: *arXiv preprint arXiv:2006.04418* (2020).

[120] Ken-ichi Funahashi et al. “Approximation of dynamical systems by continuous time recurrent neural networks”. In: *Neural networks* 6.6 (1993), pp. 801–806.

[121] Ramin Hasani et al. “Liquid time-constant networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9. 2021, pp. 7657–7666.

[122] Bryan Lim et al. “Temporal fusion transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764.

[123] Ramin M Hasani et al. “Liquid time-constant recurrent neural networks as universal approximators”. In: *arXiv preprint arXiv:1811.00321* (2018).

[124] Ramin Hasani et al. “Closed-form continuous-time neural networks”. In: *Nature Machine Intelligence* 4.11 (2022), pp. 992–1003.

[125] Ramin Hasani et al. “A natural lottery ticket winner: Reinforcement learning with ordinary neural circuits”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4082–4093.

[126] Jonathan Frankle et al. “The lottery ticket hypothesis: Finding sparse, trainable neural networks”. In: *arXiv preprint arXiv:1803.03635* (2018).

[127] Makram Chahine et al. “Robust flight navigation out of distribution with liquid neural networks”. In: *Science Robotics* 8.77 (2023), eadc8892.

[128] Tsun-Hsuan Wang et al. “Interpreting neural policies with disentangled tree representations”. In: *arXiv preprint arXiv:2210.06650* (2022).

[129] Paul Tylkin et al. “Interpretable autonomous flight via compact visualizable neural circuit policies”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3265–3272.

[130] Xiaoqi Qiu et al. “Recorded recurrent deep reinforcement learning guidance laws for intercepting endoatmospheric maneuvering missiles”. In: *Defence Technology* 31 (2024), pp. 457–470.

[131] Wenwen Wang et al. “Integrated Guidance-and-Control Design for Three-Dimensional Interception Based on Deep-Reinforcement Learning”. In: *Aerospace* 10.2 (2023), p. 167.

[132] Jiagang Zhu et al. “Learning evasion strategy in pursuit-evasion by deep Q-network”. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE. 2018, pp. 67–72.

[133] Bogdan Vlahov et al. “On developing a uav pursuit-evasion policy using reinforcement learning”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 859–864.

[134] Jianfeng Ye et al. “A Pursuit Strategy for Multi-Agent Pursuit-Evasion Game via Multi-Agent Deep Deterministic Policy Gradient Algorithm”. In: *2022 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE. 2022, pp. 418–423.

[135] Efe Camci et al. “Game of drones: UAV pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning”. In: *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2016, pp. 618–625.