Energy-efficient Tracking of Mobile Audio Sources via Emergent Distributed Systems

Dimitrios Pappas

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
School of Computer Science

May, 2025

Supervisors:
Guy J. Brown
George Eleftherakis

Declaration of Authorship

I declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research. This material has not been submitted, either in whole or in part, for any other academic degree or qualification at this or any other institution.

All sources used in the preparation of this thesis have been properly acknowledged. Where material from published or unpublished work by others has been included, full attribution has been given.

I confirm that this work complies with the University of Sheffield's policies on research integrity and the avoidance of plagiarism.

Abstract

Tracking multiple mobile audio sources in acoustic scenes where the layout, targets, and requirements change rapidly is a fundamental problem in the research field of tracking only through listening with computational means. Recent developments in the field of robotics and artificial intelligence have enabled researchers to further the capabilities of such systems – interconnected or not – towards solving the localisation and tracking problem. Nonetheless, such research focuses primarily on managing the accuracy of such systems with little care for energy (i.e. battery) efficiency, especially in applications where movement is required. Meanwhile, highly dynamic acoustic scenes are not always accounted for in the designs using mobile listeners.

This thesis attempts to bridge these gaps by attempting to solve this problem with a focus on energy efficiency: reaching the targets in a timely manner conserving as much energy as possible. To achieve this goal a suitable system has been designed and implemented, while bio-inspired computing has provided the key inspiration towards developing a listening and tracking strategy that can expertly adapt to such scenarios. Established machine-learning techniques have been employed to further optimise this strategy, ultimately achieving even higher efficiency through adaptation of psychological research towards improved collaborative problem solving via emergence engineering.

The key contributions of this thesis are thus: a distributed system framework based on microservices tailored for modern devices capable of listening and tracking with both simulation and real-world deployment capabilities, an adaptive strategy that can be utilised for standalone system solutions, and an even more efficient approach for cooperative solutions. An example application could be the deployment of several small robots in disaster scenarios for reaching and aiding trapped individuals (e.g. building on fire with heavy smoke). Finally, the interdisciplinary research process followed throughout this undertaking aspires to offer an incentive for other researchers to pursue similar avenues for innovative applications or efficient solutions in the pertinent domains.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ACF	Auto-Correlation Function
API	Abstract Programming Interface
ASA	Auditory Scene Analysis
BDI	Belief – Desire – Intention
BFS	Breadth-First Search
CASA	Computational Auditory Scene Analysis
CCF	Cross-Correlation Function
CNN	Convolutional Neural Networks
CPS	Collaborative Problem Solving
DAI	Distributed Artificial Intelligence
DFS	Depth-First Search
DL	Deep Learning
DNN	Deep Neural Network
EDS	Emergent Distributed Systems
GAN	Generative Adversarial Network
HRIR	Head-Related Impulse Response
HRTF	Head-Related Transfer Function
ILD	Interaural Level Difference
IR	Impulse Response
IRL	Inverse Reinforcement Learning
ITD	Interaural Time Difference
JSON	JavaScript Object Notation
MAS	Multi-Agent System
MDP	Markov Decision Process
P2P	Peer-to-Peer
REST	REpresentational State Transfer
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SL	Supervised Learning
SNR	Signal-to-Noise Ration
SOA	Service Oriented Architecture
SSL	Sound Source Localisation
TDoA	Time Difference of Arrival
UL	Unsupervised Learning

Chapter 1

Introduction

Distributed infrastructures have become an integral aspect of our daily lives, whether as cloud platforms, social networks, or the Internet itself, when perceived from a higher level of abstraction. The technological breakthroughs of the last decade regarding such capabilities, of what are now known as smart devices, such as their energy reserves and processing potential, enable an assortment of applications that can solve complex problems beyond merely promoting ease of offering varied services. The intricacy of such networks, however, dictates considerable effort focused on optimising and configuring them to carry out such demanding tasks in an ideal manner, an example encountered in optimal power administration on mobile robots operating in tandem to solve a specific complex problem.

Nature displays a vast array of intricate networks that have evolved across millennia, where organisms modify their own designs to adapt to environmental changes and accomplish their goals – primarily survival – with utmost proficiency for their species. Emergence is a field that delves into analysing such intricate systems found in nature and attempts to mimic their microscopic mannerisms and intelligence, whereby optimised strategies can eventually emerge at a macroscopic level. As a field inspired by the blueprints encountered in biology, emergence is well-suited for interdisciplinary research with disciplines that also draw from nature's designs. One prominent discipline pertains to reconstructing auditory scenes via computational means, which model processes of the ear and brain. This reconstruction process refers to that of interpreting and organising the mixture of sounds in an environment to make sense of what is happening. This concept is central to how humans and machines process complex auditory input, like distinguishing different voices in a noisy room or identifying sounds in nature.

One field concerned with the reconstruction of auditory scenes is the Auditory Scene Analysis (ASA), an ever-evolving field of study that focuses on how biological beings perceive an acoustic scene, primarily by the brain grouping and separating different sound sources based on cues like pitch, timing, and spatial location [2], spurned by the speculations around the cocktail party effect [3].

Computational ASA (CASA) is a related field that focuses on how machines can be infused with the capacity for performing the exceptionally complex processes that biological entities, primarily humans, can perform [1], such as localisation of a speaker using two ears or distinguishing between a musical instrument and a person speaking. CASA has tremendous potential to address real problems, from optimising hearing aids to separating individual instruments in a song. The latter, for example, can help music production fix issues with a specific instrument recording, or be analogous to the case of having different humans speaking and needing to isolate the speech of only one individual instead.

Putting together multiple systems with these capabilities could increase the potential for solving problems of a much larger scale, especially when there is interaction among them. Past work with decentralised systems and insights from biology motivated the attempts of this project to discover commonalities between fields, with a view to harnessing the best of each to develop a solution to a complex problem. Moreover, it evoked the aspiration for the project to become a framework for further evolving the results in the future, or for applying it to other neighbouring disciplines and families of problems due to its multifaceted and convertible design.

As will be elucidated during this thesis, there is demonstrated capacity for research merging social behaviours, animal hearing, and Emergent Distributed Systems (EDS) to produce highly optimised solutions adapting to complex auditory challenges. EDS are distributed systems capitalising on interaction between nodes modelled after social behaviours to optimise performance, which can find many applications in the field at hand. By exploring animal models capable of effectively tracking mobile sound sources, an area of research that has been found to be of high interest in the literature review and especially when optimisation problems are studied, this thesis launches on a trajectory to optimise such abilities. Moving forward, literature analysis will delve deeper into defining emergence and assessing its resulting effects, investigating bio-inspired solutions to auditory issues with a research gap and identifying computational methods best aligned with these aims. Consequently, this endeavour aspires to unite the arenas of emergence through decentralised frameworks and the computational investigation of acoustic scenarios, by exploring the least-travelled routes of the latter through the capable means of the former.

Research in the field has focused heavily on improving the accuracy and capabilities of important aspects of CASA, such as source separation and audio source localisation, or the even more recent successes in understanding speech. Recent advances in the fields of robotics, networking and smart devices have enabled the deployment of CASA systems in such a manner that they can solve more complex acoustic problems and inspire innovative applications. Among all research fields in CASA, studies will be presented that highlight the lack of focus on moving listeners, and capitalising on traits biological entities other than humans exhibit. And even those approaches tend to disregard the aspect of energy available to the device performing the task, which can be crucial in specific scenarios involving autonomous operation and a race against time.

Consequently, the cornerstone of this research is to design an infrastructure

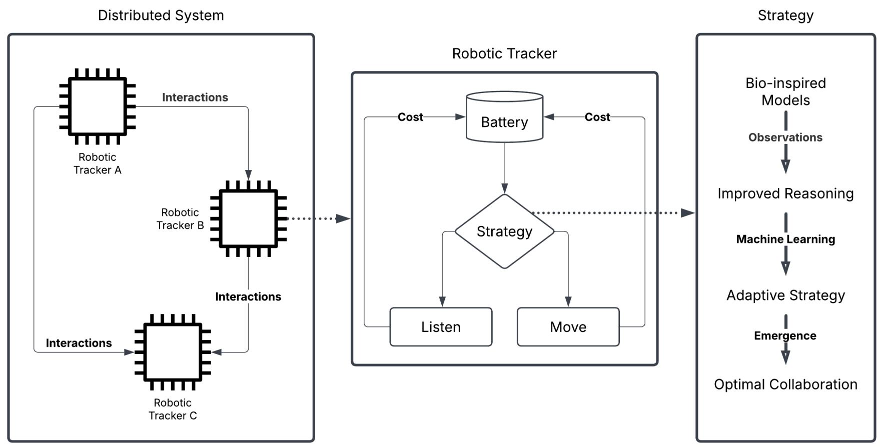

Figure 1.1: Illustration of the expected outcome of this study: a distributed system of intelligent devices, which can solve the problem via optimal strategies that have evolved through machine learning from bio-inspired models and emergent interactions (e.g., sharing, inquiring information).

for autonomous smart devices that can solve the tracking of targets of interest (sound sources) in dynamic environments, with a focus on minimising both time and energy spent to do so, without any visual aids and reliant on sound alone. Avoidance of visual aids is key in the application area this work can find, such as disaster scenarios (e.g. earthquake, fires) with reduced or no visual capabilities allowed. This is achieved with the individual nodes of a distributed system possessing Artificial Intelligence (AI) faculties, driven by basic Reinforcement Learning (RL), and augmented by socio-cognitive interaction traits towards Collaborative Problem Solving (CPS), that is a process where multiple individuals or groups work together, combining their knowledge, skills, and perspectives to identify, analyse, and develop solutions to complex problems. The contributions of its results could be the facilitation of innovative and significant applications in fields such as emergency response to disaster scenarios (e.g., rescuing people in a building on fire, helping people trapped beneath earthquake ruins).

The solution to this problem developed through this thesis attempts to capitalise on advancements in the field of software engineering for the system design, in addition to adopting bio-inspired decision-making based on the capacity of treefrogs to track their targets efficiently be hearing alone, owing to the unique energy-saving approaches they have adopted over years of evolution in their hostile environments. Arboreal locomotion, which the treefrogs are strongly characterised by, ordains that the subject returns to its arboreal state as soon as possible, spending as little time and energy as possible to achieve any goals, following nature's principle for the conservation of energy [4]. Nature's evolution has made treefrogs capable of achieving this during their mating season, where they descend from the trees and the females attempt to find the males [5]. The strategies they employ to do so vary and can serve as inspiration.

Biological research related to treefrog behaviours will be discussed that highlights their capacity for efficiency and performing localisation tasks as good as humans, providing an ideal bio-inspired model to adopt. The research questions pertinent to realising these goals are elaborated on the relevant piece of this work and explored through the scientific method twice: first to develop a solid such strategy for the individual, and then to determine how emergence can contribute to addressing this issue through CPS. This will be achieved through simulating tracking problem scenarios as accurately as possible and using bio-inspired models as a starting point to evolve optimal strategies for this endeavour through machine learning designed for capitalising on CPS scenarios. A high-level illustration of this concept can be found in Figure 1.1.

1.1 Aims and Objectives

To realise this vision of an EDS that can solve complex CASA problems in an energy-efficient manner several distinct steps must be taken. First, there needs to be a distributed system architecture, so that interaction among its subsystems, referred to as nodes, is facilitated, to lay the ground for having emergent phenomena through these interactions. Afterwards, these nodes must be capable of exhibiting bio-inspired behaviour, which can be modelled at will, as well as modified at runtime; and that is in addition to being capable of performing CASA-related functions. The next step is to put to the test several bio-inspired strategies for tracking and managing energy, with the goal of developing a strategy that can adapt to the environment and the state of the tracker to achieve its goals. Finally, social interactions among the trackers are introduced to gauge how better results in CPS scenarios involving energy-efficient tracking can be achieved.

Consequently, the higher-level aims, whilst doubling as core contributions of this thesis, can be summarised as:

A1 Implement a modern EDS for smart devices that can perform CASA functions.

A2 Utilise bio-inspired tracking strategies to develop an adaptive, energy-efficient strategy.

A3 Harness emergent behaviour through AI social interaction for CPS cases.

Each of the aims outlined above can be broken down to more atomic objectives that need to be achieved to realise them. For **A1** both a proper architecture that can accommodate all future features of the desired system, as well as a system that has the potential to be deployed and operate on modern smart devices, are essential. Meanwhile, the AI subsystems for **A1** need to be modelled in an extensible manner after the characteristics and behaviour of real-world animals that exhibit the desired qualities for solving such problems. At the same time, a robust CASA implementation needs to be utilised for locating audio sources so that the smart devices equipped with the system can track them. With regards to the bio-inspired strategies for **A2**, there needs to be a faithful implementation of the original strategies, with verification of their properties, followed by iterative experimentation of introducing new behaviours and evaluating their outcomes, leading to the formulation of the coveted strategy. Finally, **A3** demands introduction of social and relevant characteristics, followed by similar iterative experimentations as with A3 to determine the effects of the new behaviours and to what extend they can contribute to CPS.

The objectives, as relating to the aims, are thus defined as:

A1-O1 Design and implement a modern distributed system for smart devices that can support emergence.

A1-O2 Contrive a robust sound localisation mechanism for the intelligent nodes.

A1-O3 Instrument the AI to model the bio-inspired behaviour.

A2-O1 Evaluate the performance of original bio-inspired tracking strategies.

A2-O2 Utilise the findings from bio-inspired strategies to develop a new, artificial one and evaluate performance.

A2-O3 Develop the adaptive strategy for energy-efficient tracking through RL.

A3-O1 Introduce socio-cognitive traits and behaviour to the AI nodes.

A3-O2 Evaluate the performance of the evolved strategy in CPS attempts and its capacity for emergence.

1.2 Research Project Overview

Apart from the current introductory chapter and the conclusions to this thesis, there are 4 core chapters to showcase the work and present the findings of this research endeavour. The next chapter presents an exploration of the field domains explored by this research effort, focusing on the study of the fields pertinent to the topic. The trailing three chapters break down the core work with respect to the aims into thematic segments, each one also containing a discussion on recent advances, theory, and technologies apropos of the work they focus on.

An overview of the interdisciplinary research fields The primary goal of this review was to identify and study the domains with the goal of understanding their inner workings thoroughly, whilst pinpointing their possible interactions. The secondary goal was that of discovering the research opportunities. This chapter searches deeper through the fields of biological hearing, for humans and animals, discussing ASA and CASA, how they can be implemented and their potential applications. Autonomous distributed systems based on the concepts of collaborating intelligent sub-systems are explored, with a focus on sensor networks analogous to listening nodes. Emergent distributed systems are studied next, presenting a case of exemplary work on the field that can be refashioned into the desired system. The summary of this section ties everything together to showcase the narrative of the current project, as it was conceptualised and how it evolved into its final form.

A distributed system for tracking mobile audio sources (A1) To start with, a deep dive into the architecture of the expected system is presented. The reasoning behind the frameworks and technologies chosen is explained, and how a simple mobile app or a low-capabilities device can be utilised as a node in the system. What components is the system comprised of, and all the modes of interaction that they can have with each other, as well as how they can be mixed on assorted devices to formulate extremely versatile systems that can be applied to a vast array of scenarios, is also discussed. Additionally, this chapter includes the details of the CASA subsystem included in the nodes that can perform tracking tasks. The process for ensuring that audio source localisation can take place while being is as accurate as possible is described, and how this subsystem interfaces with the nodes. The use of a supercomputer for generating the required artefacts for the simulated sounds is also delivered.

Development of an adaptive, energy-efficient tracking strategy (A2) this thesis next presents the bio-inspired approach to solving the problem of tracking mobile audio sources efficiently. Implementation of the different strategies is the core topic, both the original ones and the derived ones that could compete with them. The chapters focus on detailing the environment, the actors, and the methodology for the experiments that take place to evaluate the performance of the strategies in an iterative manner. It also presents how the design of the experiments evolved in a stepwise manner during the development process. The research methodology to this end is outlined, as well as the methodology for exploiting the results towards developing the eventual adaptive strategy. Research explored explains the choice of reinforcement learning over other solutions. The chapter concludes with a discussion on the findings of the experiments carried out to achieve the research aims related to this work.

Effects of emergence on collaborative tracking (A3) The final chapter focuses on enabling the system to have more intricate and meaningful emergent interactions. While emergent phenomena arise event from the simpler interactions of the work carried out for the previous chapter, merely by means of having intelligent systems interact with the environment and their target, the different traits that can be associated with interactions among intelligent entities can lead to more interesting results. Consequently, the investigation of socio-cognitive traits that have been found to affect CPS capabilities in humans are incorporated in the system and put to the test, in a manner like the preceding chapter. The discussion of the findings ends the chapter and attempts to also convey the overall impact of the work and its potential.

Chapter 2

Literature Review

2.1 Introduction

This section revolves around a high-level, preliminary review of literature that was carried out for the interdisciplinary research fields pertinent to the concept of this thesis. These fields are, summarily: the listening through both biological and machine-facilitated means, the distributed systems infused with intelligence, and emergence in such systems. Primarily based on exploring the basics of the background theory of each area, it also attempts to explore research opportunities to capitalise on. This literature review specifically focused on the earliest stages of this research, with a view to gaining a better understanding of the fields.

In a stepwise manner, the aims of this review are to:

1. Gain a better understanding of *biological listening*, for both humans and animals, highlighting treefrogs as models.
2. Explore the realm of *machine listening* for techniques to be used and showcase the research gaps pertinent to this study.
3. Present the concepts of *autonomous distributed systems* that could be used towards realising the goal of the study.
4. Research *intelligent agents* and determine which architectures could be beneficial for each phase of this project.
5. Define *emergence* and its potential contributions through the lens of apropos work in computing and social sciences.
6. Summarise the findings about useful architectures and research gaps that this study focuses on to achieve its goals.

2.2 Biological Listening

This section focuses on presenting some details on how humans hear, touching on the surface and fundamentals of the research field that governs these important functions, while expanding to animals and bio-inspired approaches, too. It concludes with a presentation of treefrogs as the primary source of inspiration for looking to solutions for the problem at hand, highlighting their strengths towards tracking efficiency that could be leveraged.

2.2.1 Introduction to human hearing

Hearing, or audition, belongs to one of the traditional five senses and unlike smell, taste and sight, which are based on chemical reactions, it is similar to touch in that it requires sensitivity to mechanical stimuli [6]. The external, mechanical stimulus in the case of human hearing is sound, which can be heard through solid, liquid and gaseous media. Sound can be detected via the mechanical phenomenon of vibrations, where variations in the pressure of the surrounding matter can be perceived through the ear. This indicates that the process of auditory perception is a primarily mechanical process, and it has been described as mechanosensation, due to the hearing functions that the hair cells residing within the ear possess [7].

Hair cells alone do not account for the whole of the peripheral auditory system, in fact they are present only in the innermost region of the ear. The human ear can be broken down to three regions, the outer, middle and inner ear, each of which performs a different function. Therefore, hearing is a multi-step process and as such the auditory system is composed of several subsystems that work in tandem. This includes the sensory organs, the ears, and parts of the human sensory system, the nerves that pertain to audition. Consequently, the purpose of the auditory system is transduction, in essence the conversion of energy from the mechanical waves of sound into nerve impulses understandable by the human brain.

Outer ear The outer consists of two parts, the pinna, which refers to the visible parts including ear lobes and concha, and the auditory meatus, the ear canal, which serves as the passageway to the rest of the parts of the ear. The purpose of the outer ear is thus to gather energy from the mechanical waves of sound and funnel it to the eardrum in the middle ear. Sound waves are reflected but also attenuated when they encounter the pinna, a property that can be attributed to its vertical asymmetry; modifications to these waves carry cues along the ear canal that ultimately enable brain to perform several useful functions, such as determining the vertical direction that the sound came from [8]. The sound waves enter the auditory meatus after hitting the pinna and the external ear selectively boosts sound pressure with frequency in the range 2–5kHz from 30 to even 100 times more, while the ear canal itself is capable of amplifying sounds with frequencies in the range of 2–4kHz in adults [9].

Middle ear In the mechanosensation process of hearing performed by the auditory system, the middle ear plays a pivotal role. This is attributed to the fact that it is capable of converting the vibrations from the sound waves, essentially the changes to pressure in air, into fluid medium disturbances, those found in the inner ear [10]. Consequently, the mechanisms that the middle ear provides transfer sound energy from one medium type to another efficiently. The waves are carried by the ear ossicles, a series of three consecutive bones, the malleus, the incus and the stapes [11]. These delicate bones leverage the low-pressure eardrum vibrations and turn them into high-pressure vibrations for another membrane down the path, the oval window, which separates the air medium in the ear (middle ear) from the water medium (inner ear) [12]. The overall machinations of the middle ear achieve another goal: they account for impedance matching and thus maximise energy transfer so that all information is still carried through in wave form [11].

Inner ear The work of the inner ear begins when the sound waves reach and pass through the oval window in the middle ear, now in turn encountering the fluids within the cochlea, a spiral-shaped tube. In its length, the cochlea is divided in two by the organ of Corti and within it the basilar membrane vibrates when sound waves carry from the middle ear and through the cochlear fluid. This membrane is responsible for the transduction of energy from mechanical to neural that takes place within the inner ear; specifically it is made possible by the presence of over $32,000$ hair cells and their depolarisation by the movement of the basilar membrane [13]. While the hair cells do not produce action potentials on their own, the emission of neurotransmitters at the synapses with fibres of the auditory nerve trigger spatial and temporal firing patterns and achieve the transmission of sound information to the brain [14].

Another integral property of hearing, which is realised through the inner ear, is the spectral separation of sounds. The basilar membrane has tonotopic properties, therefore it is capable of spatially arranging where sounds of varying frequencies are processed; in particular, higher frequencies are processed closer to the entrance of the cochlea, whereas the lower frequencies are processed closer to its apical end [15]. This also signifies that the cochlea of the inner ear has the capabilities of a frequency analyser, as well as the abilities of a non-linear acoustic amplifier [8].

2.2.2 Auditory scene analysis

The purpose of the auditory system is to capture the sounds from all the sources and then convert them into signals perceivable by the auditory nerve. It has several astonishing capabilities, such as spatial and spectral frequency separation, identifying direction of sound sources and amplifying sounds waves. All these abilities are brought together to assist the brain in determining all important information, to analyse all sources of sound and understand the auditory scene around them. Auditory Scene Analysis (ASA) is a term coined by psychologist Albert Bregman who proposed a model that can serve as the basis of human auditory perception [2]. The core idea behind the model is to provide a process detailing how the auditory system manages to organise the perceived sounds into elements meaningful for the human brain.

An auditory scene may be teeming with assorted acoustic sources that produce sounds composed of numerous interconnected parts, yet what arrives at the ear is a single signal. For this complicated, incoming sound signal to be understood, it is imperative for the auditory system to be able to recognise the individual sound patterns. This is achieved by creating partitions and distributing bits of auditory information (i.e. unique characteristics of a specific signal) accordingly to portions that describe individual sounds. Consequently, ASA relies on the process of grouping, segregating and integrating sensory information to form the auditory streams, distinct mental representations of the sounds in the scene [16].

In the previous section, the peripheral auditory system has been analysed primarily, but the central part in the brain plays an important role in ASA, too. ASA is a complex cognitive process that allows the brain to interpret and organise the multitude of sounds in an environment. Much of this processing occurs in the central auditory system, particularly in the auditory cortex, which is in the temporal lobe [17]. The brain takes in raw auditory input from the ears and then uses various mechanisms, such as grouping, segregation, and recognition, to make sense of the soundscape [18]. Naturally, these processes assist the listener in distinguishing individual sound sources, such as separating speech from background noise or identifying the melody within a song.

Grouping and segregation occur as the brain identifies patterns in sound, using cues like pitch, timing, and spatial location to determine which elements belong together [17, 2]. For instance, if multiple sounds originate from the same location and share similar frequency characteristics, they are perceived as a single auditory object [18]. On the other hand, if sounds differ significantly in these aspects, the brain segregates them into separate streams and, once the sounds are separated, recognition involves comparing them to stored auditory memories, thus allowing the brain to identify familiar voices, instruments, or environmental sounds [19]. This entire process is supported by interactions between the primary auditory cortex, superior temporal gyrus, and prefrontal cortex, ensuring that auditory information is processed efficiently for communication, navigation, and environmental awareness [20].

2.2.3 Bottom-up grouping in ASA

The two ASA terms for the two methods used for grouping are simultaneous and sequential grouping respectively [2]. The purpose of sequential grouping is to perceptually fuse over time what would appear to be matching sounds. Experiments have been conducted that indicate any type of grouping follows the same principles of similar phenomena described by Gestalt psychology regarding vision, where perception forms clusters of matching sensory inputs [21]. In the case of the auditory system, this can be attributed to frequency variations in the sound wave and results in streaming, the perceptual segregation of constituents of the sound wave in separate streams that are meant to represent a distinct environmental sound pattern (e.g., speech, dog barking, musical instrument) [2].

Sequential grouping serves the purpose of associating sounds across time, while simultaneous grouping assists in associating sounds concurrently. Essentially, this enables the recognition of the different sound sources in an acoustic scene, achieved by integrating grouping cues that operate across frequency. An example of such a cue can be that the frequency is an integer multiple of a common fundamental, which is the case for the periodic class of sounds (e.g., musical instruments, animal calls) [2]. Using such cues, the auditory system manages to separate the sound sources, distinguishing between an animal call or a human voice, or between two different human voices.

Experiments with a sound wave containing alternating tones of different frequencies have been conducted to determine how segregation is performed [22]. A higher difference between the frequencies of the tones results in the formation of two streams, one where for the lower ones and one for the higher ones, whereas a smaller difference results in only one stream perceivable by the brain. Moreover, it appears that intermediate differences in frequencies may result in either a single or a couple of auditory streams, and it might even be possible to switch from one to the other with effort or under certain conditions.

The presentation rate also appears to affect the streaming phenomenon, as do the spatial direction and acoustic transitions. A higher rate results in less ambiguity even with smaller frequency differences, which could be attributed to a type of "recent memory" kept by the primitive mechanisms of the auditory system [2]. This could also influence the sequential grouping in acoustic transitions, where it has been observed that gradual changes to a sound towards matching a second sound can be perceived as a single stream, whereas the abrupt change tends to be perceived initially as a difference one. These examples constitute but a few of the grouping cues that can be exploited by the auditory system. According to the ASA model proposed by Bregman [2], as well as work based around it [23, 24, 25, 26, 3], several methods have been identified that can be relied upon to explain grouping as far as time and frequency are concerned.

Methods used for simultaneous grouping are [24]:

Onset - offset Frequency components that share an onset time are more likely to be integrated, while the same also happens when sharing an offset time, albeit to a lesser extent.

Periodicity Acoustic components with frequencies that are integer multiples of a fundamental frequency tend to be integrated in a single stream.

Spatial location Acoustic components with a common origin in space tend to be grouped together, although this is possibly a secondary mechanism [24]; this could be the case due to the capability of the auditory system in humans to separate sound sources that originate from the same location.

Methods for sequential grouping are [27]:

Transition A smooth acoustic transition or a continuous one result in segregation of the sounds in a single acoustic stream.

Time - frequency proximity The streaming phenomenon indicates that components with similar frequency or those that are close in time tend to become part of the same acoustic stream during segregation.

Amplitude - frequency modulation Frequency components with the same temporal modulation are usually grouped together into the same acoustic stream.

Rhythm In the case of events that occur in separate time periods, tones that are rhythmically related are more likely to be grouped together during segregation.

2.2.4 Top-down grouping in ASA

In ASA, top-down acoustic cues for grouping mechanisms come from cognitive processes like attention, memory, and prior experience, which help organise and interpret complex soundscapes [17]. Unlike the bottom-up cues discussed above, which rely on raw auditory features (e.g., pitch, timbre, onset), these top-down cues involve higher-level brain functions influencing how sounds are grouped [28].

Prominent such top-down cues as mentioned in [28] are:

Attention and Expectation The brain actively focuses on certain sounds based on expectations, an example from the cocktail party problem being anticipating a speaker's voice in a crowded room. Selective attention allows grouping of relevant sounds while filtering out irrelevant ones, such as when tuning in to a conversation despite background noise.

Memory and Learned Patterns Familiarity with sounds, such as recognising a friend's or family member's voice, aids in grouping based on past experiences. The brain uses learned schemas of speech, music, or environmental sounds to predict and organise auditory input.

Contextual Understanding The meaning and relevance of a sound influence how it is grouped. An example would be recognising words in a foreign language, which tends to become easier with exposure. Speech segmentation benefits from language knowledge, thereby grouping syllables into meaningful words for the listener.

Cross-Modal Integration Visual cues like lip movements help group auditory signals (e.g. the McGurk effect in speech perception [29]). Gestures and body language can also reinforce auditory groupings in communication.

Cognitive Scene Parsing The brain constructs a coherent auditory scene by using logic and inference. Top-down modulation from the prefrontal cortex influences how the auditory system prioritises and organises sound streams.

2.2.5 ASA inspired from animals

While communication problems abound in human interactions in our everyday life, these problems are not limited to us alone but also extend to other animals that share our unique capabilities. Generally, the perceptual organisation of sound is performed by segregation and grouping of sounds in all of the vertebrates capable of hearing, including categories of animals such as the goldfish, the starlings and assorted mammals [30]. Such animals are capable of segregating and grouping sounds belonging to different sources within a dynamic acoustic scene, spatially and temporally, each one doing so naturally within their own capabilities and having evolved to adapt to the intricacies of their environment. Consequently, inspiration can be drawn from the assorted species that have been studied over the years and follow their approach, when faced with a difficult ASA problem that falls within those domains.

Some animals have demonstrated exceptional capabilities with regards to what they can do and understand by even the simplest communication calls, such as a parent or child in a king penguin colony being able to distinguish the familiar voice amidst the extremely noisy chorus of the colony [31]. In fact, penguins are also capable of far more impressive feats, such as understanding to which extend the wind interferes with their calls, with adequate precision, and hence adjusting their voice appropriately to efficiently overcome the noise [32]. Penguins are also social animals, with varying types of calls that can be used when modelling a solution for a problem that could benefit from such functionalities (e.g., locating a source, avoiding environments threats).

Another example is the bat and studies performed over its echolocation capabilities, where the experiments have strongly indicated that they can analyse an acoustic scene. This is possible due to the demonstrated capacity of perceptual organisation of the echoes gathered as they fly through the area, which they form into echo streams belonging to the individual sources and objects within an auditory scene and can hence track them spatially [33]. Applications with a focus on tracking capabilities following similar techniques could be inspired by such innovative implementations. Meanwhile, starlings have been able to discriminate a target song when mixed with background noise and up to four different songs, something human listeners have not been able to do even after extensive training [34].

The treefrogs are also another family of animals that are excellent listeners, expertly capable of tracking sounds in highly noisy environments during their mating attempts, due to their capacity for localising sounds within just a few degrees of error [35]. Drawing inspiration from their behaviour and unique capabilities, complex communication problems that are analogous to the cocktail party problem can be solved efficiently in an innovative manner [36, 37]. Not all animals should be eligible for study and modelling, rather best effort should focus on those that are biologically and evolutionary close to humans (e.g. two ears – localisation), but also those that exhibit social behaviours that would mimic a cocktail party problem in their natural environment [37].

2.2.6 Proposed bio-inspired models: treefrogs

Experiments with treefrogs by animal behaviourists have been running for several decades, and the results have indicated that they tend to be experts at sound localisation thanks to their binaural auditory system [38, 39] – they can segregate sounds and group them in the correct individual sources with the ease other binaural mammals exhibit [38, 40]. Therefore, they tend to be excellent candidates for study and modelling when attempting to solve issues pertaining to the cocktail party problem [35, 41]. The important aspect of treefrog communication is that the studies over the years have also outlined the relationships between their audition functions and their social behaviour [42, 43], which is of prime importance to bio-inspired studies as it fosters emergence. Finally, studies have identified several details of the perception aspect of the treefrog auditory system, such as acoustic cues of importance and preferences on call envelope over minor cues, among others [40, 43, 5].

Treefrogs are a peculiar species of frogs that have developed arboreal locomotion, which refers to the property of certain animals to tend to climb to trees and spent large spans of their life on them [44, 45]. Most such animals do that only for specific purposes and limited time, although treefrogs fall to the category of almost exclusively arboreal [46]. This implies that treefrogs do not move a lot, which is an integral part of the CASA problem being researched and thereby are not the best candidates to solve this problem. Contrary to that assumption, the act of mating is what spurns them into action and demands of them to be extremely efficient, provided they descent down to this alien for them environment and have to find a partner as quickly as possible before their energy reserves are depleted and they need to return to their default arboreal state [46, 42].

Mating season is the most crucial part of the life-cycle in a treefrog community, as the performance of the local treefrog community can be essential to the survival of the treefrog population in a bio-diverse environment [43]. This is due to the fact that several larger frog species (e.g. bullfrogs) tend to invade the environment of a community comprised of smaller species, producing stronger sounds that tend to mask the mating calls of the latter [47]. This is more than other noisy species that are frequently present in their environs, such as crickets and birds. In consequence, if the local community cannot adapt to this increasing population or invasion by improving their mating process, cases have been observed where the whole population of the lesser species has been extinct [47]. This problem they are tasks to solve presents an environment that fosters adaptation and evolution, naturally, which are key characteristics of emergence and optimal systems.

Female treefrogs face the core problem of localising the sounds and moving closer to a male partner and, when they are close enough to the preferred source, they tend to ignore other sources [43]; yet, they also need to tackle the problem of filtering out the noise and navigate the hostile environment riddled with high vegetation in this scenario. The auditory stream input for females is thus comprised of what could be typical environmental noise and multiple audio sources belonging to different males of their own species. As such, the females also have to decide which male is the one they have to move towards, with past experiments showing that they prefer moving towards auditory sources that exhibit higher dynamic properties rather than higher static properties of the audio source (although they cannot be discounted), and which vary depending on sub-species [5, 39].

The male species, on the other hand, have a more complicated issue to face. Apart from producing the calls and filter out noise, they must perceive the calls of other male species and find a way to produce better calls, so that females will be attracted. This can result in highly varying calls from the same frog over time, something which tends to confuse females and thus make them change directions quite frequently [43, 5, 39]. Males also share another property of females, and that is the movement [42, 48]. As such, they do not stay static and merely change their call, rather they attempt to move away from noise sources or competing males with purpose, but also randomly in hope of getting closer to a female [42].

In an experiment where you have the tracker, a female trying to locate a male, and the mobile audio source, a male moving around so that a female can listen to them better, the treefrogs have been noted to fall into two categories by virtue of how they combine the core tasks of listening and moving to achieve their goal: (a) the regular breeders, who tend to wait and listen more than moving and move in smaller distances when they do, and (b) the explosive breeders, who prefer moving more and in longer distances to listening [49, 35, 48]. This can serve as the best starting point for modelling different strategies based on the observations of biologists. Indeed, by correlating the energy spent for the tasks of listening and moving to that of mobile robots and related devices performing tracking tasks, it is evident that performance data can be acquired within a properly constructed simulation. These data can subsequently be used to guide the designs towards evolved, more efficient strategies. The findings and their ASA potential described here established the treefrogs as the prime inspiration for the bio-inspired model to use towards solving the tracking problem in an energy efficient manner.

2.3 Machine Listening

The core concept behind this section is to showcase the field of machine listening, that is how we make devices capable of understanding a scene by means of sound alone. The typical architectures and tools that will also be used in this study are discussed with their strengths in mind, and thus the reason why they were used. Additionally, the advancements in the field and potential applications are also presented while looking for the research gaps that this study aspires to bridge.

2.3.1 Computational Auditory Scene Analysis

Over the years, machines have been used across multiple disciplines to solve difficult problems, or even to combine them and explore solutions to more complex problems. ASA in biological organisms is something that is handled by the central nervous system, but in practice in the research field it is a combination of analysing sound waves based on observation of biological capabilities of the organisms in question. Techniques of ASA can be performed computationally to utilise the processing power offered by one or multiple machines when analysing a sound wave. Additionally, this opens a wide range of applications, such as hearing aids or speech recognition in noisy environments. Consequently, the idea of utilising machines to perform grouping and segregation of sounds just like humans do gave rise to the Computational Auditory Scene Analysis (CASA) field [1].

The core principle of CASA is based around the idea of modelling the exact same functionality, or a close approximation to it, of the auditory system. The functions of the outer, middle and inner ear, based on the principles of ASA and how sounds are perceived, are to be turned into functions that can be evaluated by machines. Essentially, what is expected of such a system is to be capable to carry out specific tasks related to speech perception with as close efficiency as it is possible to a human. This implies a close connection of CASA to biologically inspired computing; indeed, CASA applications tend to focus on analysing input received from either one or two microphones (monaural and binaural respectively), which is what humans and most animals studied through ASA are capable of doing [1].

On the design layer, this close-knit relationship with such biological properties proves crucial to the idea of trying to solve a difficult problem from the perspective of a single listener; the cocktail party, for example, would easily be solved with a microphone placed at each sound source and everything fed back to the individual trying to localise and separate sounds. On the application layer, this enables developers to tackle problems such as having to work with limited equipment (e.g. isolating speech of individuals in a crowded room with only a few available microphones) or develop specialised products to solve difficult problems (e.g., single microphone on one ear providing localisation support for monaural human, audio scene reconstruction via a single microphone).

An interesting aspect of CASA is that there does not appear to be a common theory to the field [1]. Indeed, recent and extensive work on evaluating CASA systems indicates that both applications and evaluation methods appear to be rather splintered [50]. This demonstrates that practitioners and researchers in CASA do not attempt to computationally solve the ASA problem in its entirety by providing rivalling solutions to the problem itself, instead they choose to focus on determining efficiency and effectiveness of specific sub-problems. Naturally, this allows for the combination of potentially compatible and supporting processes to form a more effective final solution. Some methods focus, for example, on realistic sounds in natural environments [51, 52], others on developing algorithms to simulate ASA experiments of a behaviour nature [53, 54, 55]. The

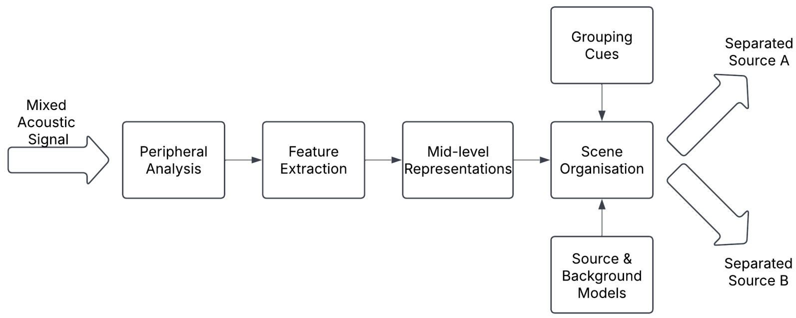

Figure 2.1: Typical architecture for a CASA system as described in [1].

core ideas behind the approaches, the architectures of such systems and the bio-inspired scene will be investigated in the following sections.

2.3.2 Typical CASA architectures

CASA focuses on analysing a digitally recorded acoustic signal and extracting meaningful, perceivable auditory information from it. Following the ASA models proposed by Bregman, machines ought to be able and receive the input, processing much in the same manner as the auditory system: perform functions of the outer, middle and inner ear. Complete CASA systems tend to follow a process where several tools are used to mimic each part of the auditory systems – if incomplete, they just attempt to solve the problems of a single step instead and optimise for it. Such systems also differ on the capabilities or approach at each individual step, but also on the point of view towards the problem, as well as the nature of the processes being employed [50].

One detailed and cohesive approach to this step-by-step process has been described, with the following core steps, in [1] (illustrated in Figure 2.1):

1. Peripheral analysis of input.
2. Acoustic feature extraction.
3. Mid-level representations.
4. Acoustic scene organisation.

During the first step, the aim is to recreate an accurate representation of auditory activity in the temporal domain, i.e. a time-frequency representation of the signal, with tools such as the cochleagrams [56]. The focus is on separating, or at least highlighting, the important parts of auditory activity contained in the signal. In biological – and hence in machine – hearing, the time dimension has been shown to play a key role in auditory perception; as time changes over time, changes in the energy in the signal indicate changes in frequency that can be monitored through such processes to discern crucial auditory activity [6].

Once that has been performed, the next step would be to attempt and extract all possible acoustic features that can be found in the stream with the help of the temporal domain representation. Such features have been discussed earlier for the ASA approaches towards managing sequential and simultaneous grouping of different sounds within an audio stream. In CASA as well, these features that can be employed include onset and offset times, amplitude and frequency modulation, as well as periodicity, among others.

Mid-level representations are something that lies between the low-level representations (e.g. time-frequency artefacts of notice) and the high-level representations (e.g. audio stream of a person speaking). [1, 57]. In essence, these processes and tools use more abstract versions of lower-level processes and tools to prepare the content for the next stage of segregating and grouping auditory information by forming the basis for segment generation. Desirable properties of such representations are: the capability of sound source separation, invertibility of extracted data (e.g. re-synthesis of a specific sound), meaningful abstraction through component reduction and focus on physical attributes as opposed to algorithmic ones, as well as a confinement to the prospective physiological capabilities of the auditory system [57].

Organisation of the acoustic scene is the last step in the CASA, the one that can lead to the re-synthesis of an individual, isolated audio stream that can be evaluated, so that the efficiency and effectiveness of the CASA system can be assessed by extent. Acoustic scene organisation relies on grouping cues that have been identified over the years for the auditory system and/or models developed for specific sound sources or types of background noise [1, 24]. One important new aspect that has emerged since the work also entertains the idea that predictive models are essential to de- and re-constructing the audio scene [50]; much like the neurons in the auditory system are capable of predicting primitive regularities, so can regularities derived from representations and identified in the input stream predict potential individual sound sources [58].

2.3.3 Signal-based processes

Several means pertaining to signal manipulation and properties have been employed for most of the typical CASA systems, with research and practice on the field attempting to improve such tools, creating more efficient ones with similar ideas or combining them for more robust solutions [50]. In this section, tools will be presented briefly that can be used in the process described in 2.2 in assorted steps of the process to realise the goals of the system. Naturally, the focus of these tools is on signal manipulation, occasionally referred to as spectral CASA, to distinguish among other approaches that will be discussed later on (e.g., predictive, neural CASA).

Cochleagram One of the popular means of deriving a time-frequency representation of the audio input is the cochleagram [1], based on the idea of simulating the capabilities of the cochlea of the inner ear and the properties of hair cells [59]. Harmonic components of the audio stream can be separated by the cochlea due to its great capacity for determining pitch and frequency [60, 1, 59], thereby simulating the cochlea functionality in a computational manner can result in the identification of artefacts of importance. Acting as a simulator of the outer and middle ear, the signal is broken down to assorted frequencies that would consequently be selected by the cochlea and the hair cells as detailed in the biological hearing.

Hair cells produce spike patterns in their operation, which could naturally be translated into a spike from the impulse response generated by a cochlea modelling process. A gamma function can be used with a tone input, to generate this impulse response, based on a gammatone filterbank [1]. Such filters have been formulated over the years through extensive observation of the auditory system physiology and psychophysical responses to stimuli, thus proving excellent candidates for realising the cochlear filtering in a computational manner [61]. The Meddis hair cell model also focuses on gammatone filterbanks and mimicking of the hair cell transduction properties [62, 63]. One such example application of this process is illustrated in Figure 2.2.

Correlogram and cross-correlogram Another process used to derive important information from the auditory stream comes in attempts to emulate the pitch perception capabilities of the auditory system [64], which much like the cochleagram is a representation in the temporal domain, too. The correlogram can also be computed in the frequency domain, if discrete Fourier transform and its inverse are applied [1]. Its importance and contribution as a model to pitch perception is attributed to the capability of bringing together both resolved and unresolved harmonics [1]. Through the analysis of the figures generated from the correlogram, and specifically the position of the peaks, the perceived pitch can be estimated. The correlogram also complements the cochleagram, given that both representations (time-frequency analysis) enable CASA systems to perform more robust source separation and auditory scene analysis by utilising both spectral-temporal structure (cochleagram) and harmonic-pitch information (correlogram).

One issue of speech perception modelling comes in the form of the interaural lag of auditory information received through each ear, as the location of a sound source has different distances from each ear, thus the signal travels at different times through the ear. A solution to this problem is the cross-correlogram, which aspires to compute the difference between the left and the right input channels of the CASA system, and which is based on various physiological studies related to this problem [1]. The process of cross-correlation, consequently, focuses on the attempt at identification of similar peak patterns in the output, which would suggest the same sound segments and/or sources.

Time-frequency masks One of the core concepts of the physical auditory perception is the fact that a specific sound can become obfuscated behind a much

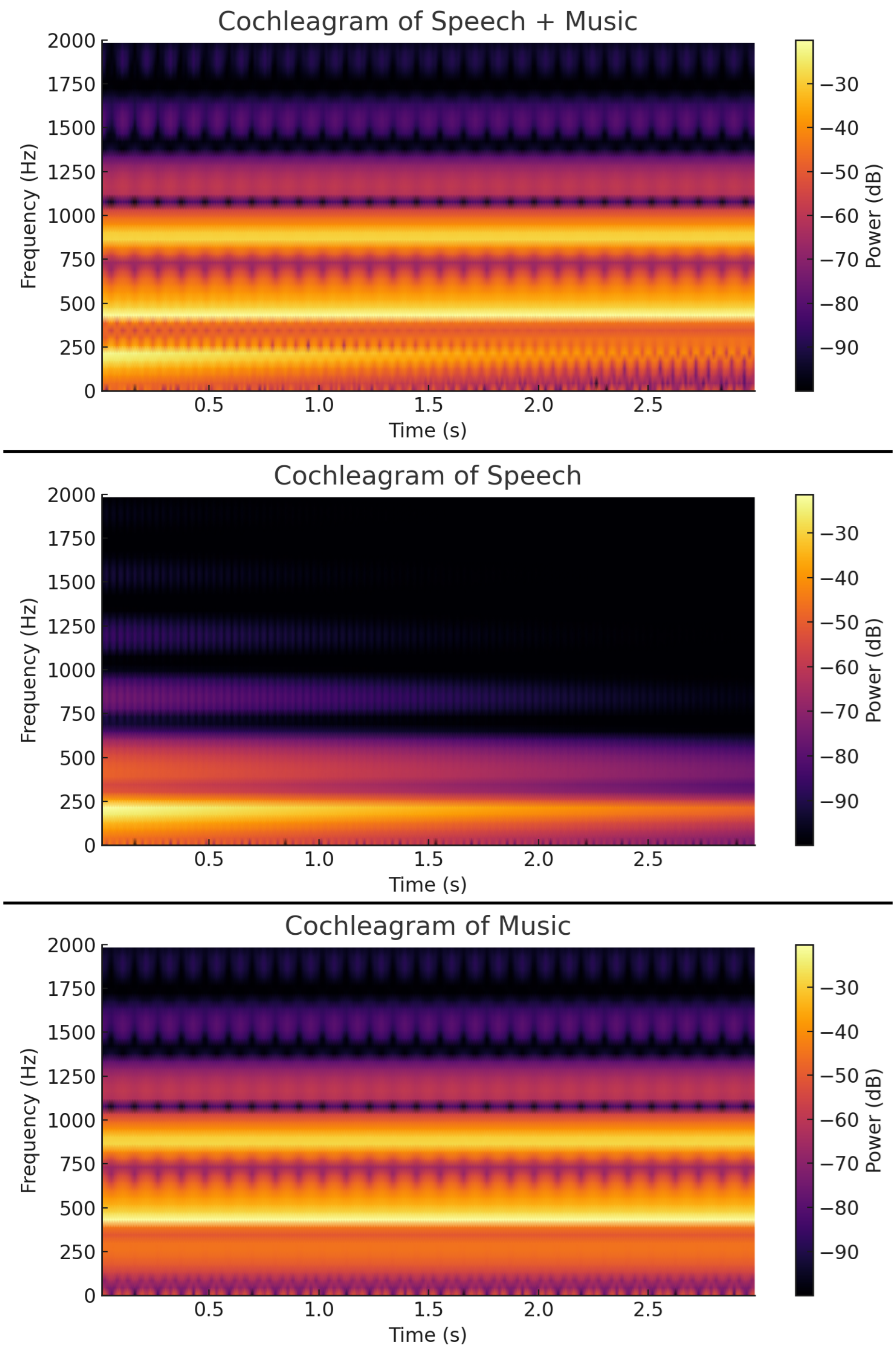

Figure 2.2: Cochleagrams showcasing the combined signal, and then the separated speech and music after the application of Meddis filtering. The frequency (Hz) on the vertical axis over time (s) on the horizontal axis. The overlap is discernible within the frequency range $0-500$ in the combined signal.

louder sound [26], which is referred to as masking and that has been considered greatly in attempts at improving speech intelligibility and perception [65, 66], and noise filtering [67, 68]. This gave rise to the idea of applying masks over lower-level processes in a CASA system, such as the cochleagram, so that the sound of desired "strength" or "level" can be segregated from an audio signal. A Wiener filter is applied to this end, which is capable of weighing the target regions and then suppressing the potentially undesired ones [1].

Resynthesis This process is straightforward in that it merely attempts to synthesise the original audio signal from a segregated source. An example of doing that can be seen using a cochleagram and the desired masks that have been applied to it. All the information required to isolate the desired sound sources and group them along time has been gained at this stage. Consequently, the masked cochleagram or correlogram can now be inverted. The masks account for the desired energy levels by weighting them, thus removing unneeded auditory information and reducing noise, and by the sum of all weighted responses emerges a reconstructed audio signal containing the desired sounds [1]. This is one of the more traditional processes, where recent advancements that will be discussed next have offered new ways to tackle such problems.

2.3.4 Advances in CASA modelling

Whether inspired by neuropsychological properties, simple physics or physiological properties of the auditory system, the core systems and processes described in the comprehensive survey of the field up until 2006 [1] tend to focus on spectral analysis for segregation, grouping, feature extraction etc. Current research still aspires to improve these tools and make them more accurate and more robust in their purpose, however several new, detailed models have emerged over the last decade with different focus on processes employed or theoretical principles regarding the CASA problem. The most important and descriptive of them tend to feature aspects of predictive processing as means to the end, meanwhile offering competing approaches to signal interpretation, but also with theory behind them that can be drawn from Bayesian inference, Neuropsychology or Temporal coherence [50].

Predictive processing has been shown to contribute immensely to the auditory system towards sound segregation. This is generally attributed to the fact that, if a representation of a certain sound segment from a specific source has been generated, then it is much easier to predict future sounds from the exact same source [58, 69, 70]. Models based on Bayesian inference, the most prevalent ones being [54, 71], concentrate largely on reaping the benefits of predictive processing. The models chronicle the acoustic scene in sound object classes or vectors that describe all its dimensions, and which can be approximated from the input. The Bayesian algorithms utilised to generate predictions take the vectors or classes as inputs and compete on the probability of occurrence for concurring vector dimensions or class properties. The goal in [54] is to produce series of discrete states of the acoustic scene, where discovered sound segments are either integrated in a single class or segregated in more classes, with successful qualitative evaluations. In [71], binaural input is provided and the model attempts to segregate one or two voices present in the signal, with evaluations showing more efficient and effective segregation with a single voice, but successful segregation of voice envelopes nonetheless.

The application of Neuropsychological and Neurophysiological findings in the field of audition to CASA has given rise to several models, which aspire to infuse representations across all levels of their architectures with properties of neurons or neural networks [50]. Characteristic attempts at this field are those of [53, 72]; the former details auditory artefacts as linked to neurons that interconnect and interact with the neurons of other auditory artefacts, whereas the latter employs representations of the various frequency channels as neural oscillators in a two-dimensional map in the temporal domain. Predictive algorithms are featured in [53], since the sound objects with the attached neurons are designed to compete and it is imperative to ensure proper segregation of future sound objects, with the end goal of produces a series of the most prevalent identified auditory artefacts. The oscillators in [72] are used to represent a single object when synchronised, different ones when desynchronised, the activation and excitement level of which is reliant on the external input. The expect results of the model are based on the synchronisation of the oscillators, therefore it can either be a state of integration or one of segregation, with results adhering to expectations for audio streaming.

One of the new modelling principles is based on temporal coherence, which is used to measure the average correlation between the value of a wave and itself delayed by a specified interval (coherence time) at any pair of times. This can demonstrate how a signal can interfere with itself across time; therefore a property that can be observed is that across a single frequency the wave is perfectly correlated with itself. Consequently, this can assist with segregating and grouping together acoustic segments belonging to the same source due to observations across time of their temporal coherence [73, 52]. Two models based on temporal coherence are [52, 74], where the former generates a spectrogram of the provided auditory signal and clusters features based on temporal coherence, and the latter follows the same principles but focuses on clustering through predicting subsequent inputs. In [52], two masks are generated by the model that can be applied to the input signal so that the segregated streams can be recreated individually, while [74] produced 5-dimensional stream representation that can be utilised to re-synthesise the spectrogram of an auditory artefact.

2.3.5 CASA with neural networks

In recent years CASA has witnessed significant advancements propelled by deep learning techniques. One notable development lies in the realm of source separation, where Deep Neural Networks (DNNs) are employed to disentangle overlapping sounds within an acoustic environment [75]. By leveraging Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [76, 77], researchers have achieved remarkable results in isolating individual sound sources from complex audio mixtures [75, 78, 79]. This capability finds applications in various domains such as speech enhancement, music transcription, localisation, and noise reduction [78, 79, 80, 75].

Moreover, deep learning has revolutionised the field of sound event detection and classification. Traditionally, this task involved handcrafted feature extraction and classification algorithms. However, deep learning models, particularly convolutional and recurrent architectures, have shown superior performance in automatically learning discriminative features from raw audio signals [80]. These models excel in detecting and categorising diverse sound events, ranging from environmental sounds like footsteps and car honks to human activities like speaking emotionally [81] or even music classification [82].

Furthermore, deep learning techniques have facilitated the integration of contextual information and spatial awareness into CASA systems [75]. By incorporating multi-modal inputs such as visual data from cameras or depth sensors, alongside audio signals, deep learning models can better understand and interpret complex auditory scenes [75, 83, 84]. Consequently, this interdisciplinary approach opens avenues for applications like autonomous vehicles, augmented reality, and smart home systems, where accurate perception of the surrounding auditory environment is crucial for decision-making and interaction.

Another exciting frontier in CASA is the synthesis of realistic auditory scenes using Generative Adversarial Networks (GANs). GANs enable the creation of synthetic audio data that closely resembles real-world recordings [85], facilitating data augmentation and training robust models [86]. This could be helpful in interdisciplinary research endeavours attempting to leverage the tools CASA provides through GANs. By training GANs on large datasets of audio recordings, researchers can generate diverse acoustic environments with varying background noise, reverberation, and spatial characteristics [86, 85]. Eventually, these synthesised scenes can contribute to the development of more robust and generalisable CASA systems.

2.3.6 Bio-inspired CASA applications

As it has been established, CASA itself is an inherently bio-inspired field. Nonetheless, biology offers several fields on its own to draw inspiration from for CASA practitioners and researchers. Inspiration from neuroscience – neurophysiology and neuropsychology – has contributed to how neural CASA systems have been designed and operate. A prime example of a bio-inspired CASA system would be the formulation of a neural network based on the capabilities of temporal coherence that aspires to improve the capacity for double-vowel segregation [87]. Drawing from the evaluation results of this system, the idea of using a simple, monaural input source and exploiting the capabilities discovered in the neural network of spiking neurons could provide better results for source separation [88]. Such capacity of an evolution of a CASA system through the exploitation of earlier results is what drives research in bio-inspired CASA. The research fields involved also span across various disciplines.

Another bio-inspired potential application could be the utilisation of CASA techniques for localising objects to assist people with blindness. Attempts in this field have combined sound systems, based on the binaural capabilities and head-related transfer function, in an augmented reality manner to help blind subjects localise sounds, proposing a device combined with image processing that will be capable of producing sounds that accurately localise the visually identified objects [89]. Researchers working on this field could also draw further inspiration from related bio-inspired ASA attempts, such as the echolocation capabilities of bats discussed earlier that have shown great promise towards localising auditory objects in a sonar-like manner.

Other attempts bring together the fields of CASA, biology and robotics towards providing a more authentic and unique interaction experience of the user with a robot. As an example, related work on the field has produced a system that is capable of synthesising animal sounds at real-time through a highly parametrised system mimicking mammal voice production [90]. Such a tool could be highly valuable for practitioners and researchers working on CASA applications using animals as the inspiration for solving or studying more complex ASA problems, if the effort is spent to create presents for the desired animal families and hence forgoing the need for recording the proper sounds from live animals.

Meanwhile, more traditional CASA applications also benefit from bio-inspiration for optimising algorithms tailored to solving specific tasks. As an example for speech recognition optimisation, modelling the system after the exact properties of an outer and inner/middle ear, and combining these behaviours with appropriate tools (Gammachirp auditory filterbank, low-pass filtering) has resulted in improved feature extraction in noisy environments [91]. Others focus on boosting monaural speech segregation capabilities by focusing on more efficient energy extraction based on bio-inspired models [92], or bio-inspired tensor representation of the input for more efficient separation of two seemingly identical noise sources [93].

The range of possibilities for CASA is vast, including even solutions inspired from other animals as mentioned previously and not only from human biology. As discovered earlier, such attempts still concentrate on improving the effectiveness of the algorithms rather than exploring more paths to solving existing problems [50]. Integrating numerous lower-scale solutions to the CASA problem to create a larger-scale solution, or even bringing together different solutions to a specific aspect, is one of the best directions for future research [50, 36]. Nonetheless, the complexity of the individual parts or that of the expected sum can be an impediment to the operational capabilities of the system, therefore great care and effort towards optimisation and testing of the end-system is required [36]. In conclusion, a gap in studying optimisation from other perspectives for applications (i.e. energy efficiency for this study) supports the choice of tacking the tracking problem in this study.

2.3.7 Localisation and tracking

In the field of CASA, audio source tracking as an abstract concept is the problem of estimating the evolution of the positions of sources over time, especially when the sources are mobile. Sound Source Localisation (SSL) is one of the core techniques used for tracking, even if it is not used for every step of the way. Indeed, there are technologies that employ SSL only for locating a potential sound source of interest, thereafter relying on other cues that can be derived from the received input in combination with expert knowledge of the sound source and the expected environment – a very limiting approach, however, for dynamic environments [75, 94, 95]. Other tracking solutions rely on more complex and elaborate techniques, combing AI and distributed approaches that will be investigated next.

To begin with, and considering the real-world problem this thesis attempts to provide a solution for, the field of robotics is where the developed system can find application in. Robotic solutions as trackers have a very specific obstacle to overcome, and that is the isolation of internal sounds, commonly referred to as self-noise or ego-noise [96]. Solutions tend to follow the predominant and established *stop-perceive-act* principle (**stop** to prevent impulsive actions, **perceive** the situation by gathering relevant information, **act** based on informed reasoning and analysis), which this thesis follows and that also lends itself to RL controlling agents, otherwise they attempt to listen continuously and tend to resort to proper CASA processing filtering techniques for noise reduction to remove the noise with varying degrees of success [96, 95, 75]. Other continuous listening solutions attempt to solve this issue by implementing inner microphones that get a near perfect representation of ego-noise for signal filtering properties [97, 95]. Admittedly, such implementations can provide better tracking (i.e. target localisation over time) precision, although they acknowledge the concerns of increased energy costs for both continuous listening and extra processing requirements [97, 75].

2.3.8 Conventional vs. AI localisation and tracking solutions

A common tracking technique for solving the problem include the computation of the Direction of Arrival (DoA) [98]. This technique operates by calculating phase and frequency at peak values at initial processing of the signal, then continuing with any necessary filtering (e.g. noise via low/high-pass FIR/Butterworth filters [99]) before performing the estimation through energy distribution on the spatial spectrum. The key performance factors for DoA are: the number, quality, and position of the listening elements, Signal-to-Noise Ratio (SNR), and Signal Coherence (i.e. the similarity of signals confusing the listener) [98]. Tracking is achieved by comparing either continuous or discrete over time results to predict the target movement.

Having multiple listening elements (e.g. two microphones) enables the Time Difference of Arrival (TDoA) technique [100], which the currently proposed system achieves through conventional binaural localisation: an inverse Fourier transform of a weighted version of the cross-power spectrum between the signals of the two ears and finding the time delay. For the case of the CASA component here, this delay is mapped to the azimuth suggested by the HRIRs of the simulated environment to maximise accuracy – in essence comparing the actual ITDs with the theoretical ones for the environment, revealing the dependency for real-world implementation on such a model. Naturally, the environment plays a key role for the efficiency of TDoA techniques, therefore key performance indicators for DoA also apply to TDoA. SNR from reverberation and ego-noise is also a problem, the latter mostly for continuous listening attempts compared to discrete ones [100]. One key advantage of TDoA that allowed its adoption for this thesis is the moderate computational cost.

Solutions for tracking using AI and, primarily supervised or unsupervised Deep Learning (DL), have been emerging over the latest decade with the advancements in the relevant fields of neural networks. With a very simple in abstract level pipeline where they extract features from multichannel input signals to develop DoA estimates, recent surveys on SSL via DL have highlighted their superior performance with regards to localisation accuracy over traditional SSL for assorted applications [75, 79]. The prohibitive factor in adopting such a solution for the problem being explored is that these systems require adequate in both size and parameters training datasets. As mentioned briefly in the previous chapter for the choice of implementation for the CASA component, the highly dynamic environment the agent will operate it requires such a complex and immense training dataset that constitutes it virtually infeasible for the scope of this thesis. The most recent systematic review of DL acoustic systems drives this point home with the results pertaining to the aspect of mobile tracking [75].

2.3.9 Distributed localisation and tracking solutions

At this point, distributed localisation and tracking solutions can be introduced. As mentioned for the conventional approaches above, increasing the number of microphones provides a new way of solving the problem. Consequently, several techniques have been utilised to deploy multiple audio sensors, either combined in an continuous array or as statically placed audio sensor networks across the area [101]. Conventional methods can be applied comparing the inputs for each sensor in an algorithm to this end, but generally such approaches capitalise on the concept of beam-forming: an acoustic energy map developed by the combined information from all the sensors being used, which is subsequently analysed with assorted techniques to decide on the most accurate direction (e.g. Steered-Response Power – SRP) [101, 78].

Distributed approaches produce excellent localisation results, but on the other hand they introduce several costs not only on physical equipment, but also in computational power – and by extent energy in autonomous devices [102, 101]. This fact demands a balance between computational costs and desired accuracy, but even beyond that the sheer majority of implementations for such research approaches are distributed only in the node deployment aspect – computation is usually performed in a centralised system that process the data [103, 102]. Tracker mobility is overlooked and when the distributed mobile trackers do take the stage, thus the computation and decisions are again delegated to a centralised intelligence that attempts to map the acoustic scene (e.g. acoustic Simultaneous Localisation and Mapping – SLAM techniques [104]) and not to reach a target in a timely manner to offer assistance [83, 84].

Acoustic SLAM research is the closest family of techniques localisation to the problem this thesis aspires to provide a solution for. SLAM is primarily used in robotics using visual aids towards mapping a scene [84, 104], whereas in acoustics sound is also added [105, 104] – or in the few distinct cases of interest to this study such as aSLAM for a room or its underwater equivalent [106], it comprises the only type of sensor. The core concept of SLAM is to localise the moving observer within its environment, while also mapping the environment itself. Acoustic SLAM implementations rely on a Bearings-Only Tracking (BOT) strategy [104], which is the process of tracking a target through the bearing measurements collected by a moving listener – akin to the *stop-perceive-act* approach but in a more continuous manner.

What makes SLAM-oriented approaches unsuitable for the problem studied herein is the purpose with which they are designed: to simply explore a place by "getting a feel" for it, rather than a razor-sharp focus of achieving a goal, with no concerns for energy cost efficiency. However, there is an application case for possible future work with acoustic SLAM architectures given a RL-based intelligent tracker: they can employ acoustic SLAM for the *exploration* functions and switching to *exploitation* actions, provided that a low-energy cost solution can be incorporated. Regrettably, even initiatives such as the LOCATA challenge and related surveys reveal the issue of a focus on static instead of moving listeners tracking the targets [107]. Recent work on near-sensor and in-sensor computing outlines the needs and challenges for transferring computation to distributed nodes from centralised architectures for energy conservation that this thesis aims to provide a framework for, too [108, 102, 103].

2.3.10 Proposed audio source localisation solution

The first choice regarding how the system will attempt to localise perceived sounds comes in the form of a decision between monaural and binaural techniques. While binaural has already been discussed in more length, monaural provides a viable alternative for localisation purposes. Monaural localisation prominently resorts to spectral analysis for discovering the cues of the signal, such as the shape (sounds above tend to have different shapes than those below), loudness (closer is louder in such analysis), or the Head-Related Transfer Functions (HRTFs) [109]. These exceptional functions are designed to determine how a sound from a point in space, parametrised as frequency and source location, would reach an ear [109]. Monaural cues are also integral for low-frequency sound localisation, but ultimately shine when robustness is needed and localisation is assisted by other sensory cues (e.g. vision) [94, 109].

Binaural techniques are employed when there is a need for higher precision in determining the location of a sound source [109, 110]. The brain of a binaural animal is capable of determining the difference in time between when a sound signal reaches each ear (Interaural Time Difference – ITD), when then assists in deciding the angle from which it originated [110, 111]. Interaural Level Difference (ILD) is also another special function performed at the brain, where the simple concept of a higher intensity sound in one ear translates into a source being closer to that ear [111]. The ILD has been found to assist in localisation in higher frequencies, usually well above $1000Hz$, whereas the ITD is much more useful for medium or lower ones. Binaural implementation can also benefit from capitalising on HRTFs operating on inputs from each ear, so the correlated result can give a much accurate estimation of the sound origin angle related to the head orientation [111].

To develop a system capable of running simulated experiments for evaluation and research purposes, the design of the experiments must be considered. At a higher abstraction level, there needs to be a specific environment where some entities can produce and others can receive sounds, and these entities can change positions as time passes thus creating and consuming sounds from different positions. Ultimately, due to the ease of providing input files in simulation, the core issue is that the function providing localisation logic to the trackers must be able to receive simulated sounds as if they were generated at a specific location and heard from a specific location.

This difficult problem can be overcome with the use of Impulse Responses (IRs), brief input signals describing the reaction of a system to some external input, which in CASA are referred to as Head-Related Impulse Responses (HRIRs) [112] – they encode the brief analysis of the monaural cues as received from the two different inputs (i.e. the two ears). Performing a convolution of an input sound wave with an HRIR produces a new one, which represents how the input sound would be heard at a specific location as originating from its source location in the environment and with the head turned towards a specific compass orientation.

A wholesome suite of tools that can cover these needs can be found in the one developed during the Two!Ears project, which aspired to develop a computational framework for modelling active exploratory listening in a manner that assigns meaning to auditory scenes [113]. Specifically, the suite provides binaural simulation for generating HRIRs by providing different parameters for a two-dimensional (2D) simulation environment, ranging from ear distance and head orientation to sound reverberation, absorption, or reflection [113]. Moreover, it provides excellent HRTF sets from KEMAR manikin measurements (a manikin designed for anthropometric research that has many applications to binaural CASA [114]) in the Spatially Oriented Format for Acoustics (SOFA as described in [115], standardised in 2015) that can be utilised by the simulator to generate the proper HRIR [113].

Evidently, using the Binaural Simulator component of the Two!Ears project can provide a large set of HRIRs to describe the auditory scene of choice for the experiments tailored to the needs of the study. The *tracking* part of the solution will be covered by the strategies that will define the intelligent nodes of the proposed EDS that will be investigated next.

2.4 Autonomous Distributed Systems

When looking for innovative solutions to complex problems, sometimes an approach where the combined capacity for intelligence or processing power in a system larger than a single device can be desirable. This section explores such systems, and specifically the family of such systems that can exhibit autonomous behaviour, which is much sought after property in a system that could be deployed in dynamic environments, to survive and overcome the problems. Means and architectures for developing such systems through multiple intelligent agents are also discussed, including bio-inspired solutions, with advancements and comparisons in sensor networks paving the way to the framework that needs to be developed for this study.

2.4.1 Autonomy in distributed systems

The term *distributed systems* refers to systems comprising of networked machines that communicate and coordinate through a common protocol, ultimately interacting with each other towards realising a common goal. Essentially, a distributed system is a collection of nodes capable of autonomous computation and that possess local memory, with the communication between them actualised via message passing facilities. The typical common goals of such systems are to distribute the workload impossible for a single node to perform to several nodes, to facilitate communication and information exchange among its users, or to govern access and delivery of shared resources, offering an impressive array of benefits to administrators and operations [116].

Distributed systems have many applications in the field of computing and problem solving, some of which were mentioned above. Some of the applications are naturally based on parallel computation, including scientific computing, where some problems require massive amounts of computations, and parallel rendering in heavy graphics applications (e.g., medical visualisation, virtual reality), or even real-time process control in systems such as an aircraft control system [117]. The advantage that such a system offers in these cases is primarily that it could be much more cost-efficient to utilise a smaller network of simple computers compared to a powerful supercomputer, to achieve the same results.

This practical benefit was the driving force behind the idea that, since these systems operate over a network, they should be used as a solution to a variety of telecommunication and network problems. Telephone and cellular communications can be enhanced by employing distributed systems [117, 118], while wireless sensor networks can achieve wider coverage and address energy-related problems when a node does not have to process the entirety of the workload alone [119]. Even the World Wide Web is essentially a distributed system in its core. Many telecommunication and networking problems can be solved much more efficiently through application of concepts borrowed from distributed systems, such as routing algorithms being enhanced through technologies like the distributed hash tables [120].

Other typical implementations include online service-oriented systems, peer-to-peer applications, distributed databases, and distributed sensor networks. Wireless sensor networks can be designed to be distributed in nature and hence span across larger areas of importance, too. Modern way of life has also found such networked systems pervading many aspects of our everyday life, giving rise to disciplines such as opportunistic sensing with multiple available devices in close proximity [121]. The sheer number of cloud infrastructures or the ubiquitous nature of distributed systems suggests challenges in the way such systems are designed, managed and operate with minimal resources [122]. The fundamental concept in distributed systems is that there is an assortment of interconnected, communicating devices with a specific goal to fulfil, and it is essential to investigate the modern approaches to such systems, as well as the much desired autonomous operation that can make them feasible [123, 124].

2.4.2 Developing autonomous systems

Autonomous operation is the most desirable property in distributed systems, expected to provide most solutions to the complex sub-parts and their management. Any preliminary attempt at research in the broader field of autonomic computing, as well as its sub-field of autonomy in the world of distributed systems, is bound to result in frustration and the formulation of a blurred concept about the definition of autonomy. Arguably, this can be blamed on the lack of uniformity in literature terms employed towards describing the concept and its aspects, ending up with a reference to the very same concepts and desired properties using several terms interchangeably: adaptivity, autonomy, self-management, self-regulation, self-organisation (leading to the often-encountered term Self-* properties) [125].

Considering all these terms, it becomes apparent that any autonomous system ought to have certain properties that enable it to operate within its intended life cycle, and all those properties have one thing in common: they are devoid of human intervention. Accordingly, the concept of autonomy in computing can be defined as the capacity of a system to operate within its life cycle to its fullest potential with minimal to virtually no human intervention at all. The concept of autonomic computing was introduced by IBM in 2001 at the beginning of the new century, expecting to see extremely rapid changes in how software is used or developed, an extremely inflated size of source code to manage and attempts at integration of heterogeneous devices and environments [126]. The proposed solutions to such issues came in the form of designing and implementing systems from the beginning that should be suitable for following several goals set by the developer, essentially administrative tasks disguised as goals, so that the systems can manage their own state as optimally as a human administrator would [127].

The literature and practice suggest that, arguably, the best solution to solving autonomy issues comes from the agent-based programming paradigm

[128, 125]. Even before the vision of autonomic computing, utilising collaborating agents manages to induce stability in the system and diminish complexity. Indeed, the agents can process the state of a distributed system and make decisions on best strategies to follow by utilising several mechanisms: rewarding right choices [129], migrating agents [130], distributed artificial intelligence [131], managing distinct system components [132], among many more. Multi-agent systems offer the most to autonomy, and distributed systems in particular, because through the properties and interactions of agents may emerge robustness, scalability and adaptability using partial views, feedback and self-evaluation functions [125]. Meanwhile, self-assembly, self-healing and self-optimisation emerge from multi-agent systems where agents are responsible for managing low level resources [133]. Another interesting approach to achieving self-organisation comes from the idea of deploying agents in a network and letting them discover resources and attempt to manage them, then offer their information or functionality as services [134].

Services and SOA are also capable of providing some degree of autonomy to distributed systems. Services can handle the task of communication or information sharing, and focusing on lightweight services with descriptions based on standards will definitely contribute to enhancing the awareness aspect of autonomy [135, 136]. Services as an abstraction mechanism can also be employed to realise agent functionality and provide access to it for other agents or sub-systems, too [137]. In addition to borrowing from the field of Service-Oriented Architectures (SOA), principles fundamental to the component-based paradigm can be utilised for autonomous distributed systems, as do software design patterns at an architectural level, such as strategy and adapter patterns achieving adaptivity and self-optimisation, or observer pattern for awareness and self-monitoring, even the abstract factory for self-repair and self-healing, to name a few [138].

The complex problem this study aspires to face and overcome is centred around the idea of a real-life scenario where several intelligent robots cooperate to track all targets by sound alone in a disaster scenario. Autonomy in this eventual distributed system is, consequently, of high value to the success. This means

2.4.3 Multi-agent systems

Multi-Agent Systems (MAS) are systems composed of many individual agents, which interact with each other to solve a problem. They are usually employed in situations where problem solving cannot be handled by a single agent, or a non-agent-based software program, much like distributed systems. The applications of MAS can range from disaster scenarios and social response modelling, to geographical information systems and logistics management [139], or shine in more complex fields, such as simulation, construction of synthetic worlds, or even robotics and mathematical modelling for complex systems [140]. Obviously, the need for many agents to cooperate basically stems from the need to distribute the entirety of the workload across many different agents, either because one agent alone cannot handle the size of the knowledge, or because it cannot solve the problem fast enough, effectively resembling distributed systems but on a localised scale.

Looking further into the machinations of a MAS, the issue of agent cooperation and life-cycle inside the environment is starting to become an important and complex issue. Indeed, in one MAS the agents may be working together to achieve the same goals, meaning that they share a purpose and each one of them tries to contribute (e.g., robot guidance), while in another MAS the agents may be competing against each other, such as the example of two stock-exchange simulation agents buying or selling the same stocks competing against each other, even if they still share the same higher purpose of taking part in trading and maximising their profits [141].

The observations naturally hint to the need of a proper design and modelling of the agents to be used, their purpose and smaller scale goals, as well as of the environment they "live" inside, too. Meanwhile, agents that face such problems during their life-cycle, and especially so when the environments they find themselves operating in are dynamic in nature (i.e. they change over time), need to be able to deal with unexpected circumstances and issues efficiently. Notwithstanding the cooperation strategies, there is a need for the designed MAS to have certain characteristics that will enable solutions to any such issues that may be encountered during its expected life-cycle, effectively enhancing its problem solving capabilities.

These essential characteristics can be condensed into [142, 143, 144]:

Autonomy The agents must be partially autonomous, adjusting as required and transferring control to humans if, and only if, it has been decided that they cannot decide. This also enables the agents to behave differently from each other, otherwise it could be argued that they work in tandem as one, effectively forming one super-agent instead that represents the undesirable central point of control, and consequently failure, inside the MAS.

Decentralisation Distribution of knowledge data and processing power to multiple agents is of paramount importance in an MAS, because it eliminates the chances of having a single point of control failure for the application, because the components of the MAS will be significantly more loosely-coupled in this case. Apparently, this also strengthens autonomy in the system, which as stated above is a much desired characteristic.

Local View Agents in the MAS should be limited only to knowledge of one part of the system only, their own part and not the whole system instead. This "part of the system" refers to the logical spatial position in the system and not actually the physical one in the network (e.g., agent relationships and not proximity of individual physical system IP addresses). Naturally, this aspect enhances their ability to achieve their goals through their localised interactions with other agents.

Utilising agent autonomy, data and processing decentralisation, as well as limiting the agent awareness to the local view, the MAS can achieve the best possible solution to the problem it was created for without substantial human intervention. A high degree of autonomy, and possibly even a limited degree, too, together with the adaptivity provided by all the other characteristics, has resulted in the MAS frequently being referred to as "self-organised systems" or "adaptive systems", as well [141]. Evidently, depending on the agent architecture, their communication and the structure of the environment, the MAS exhibit several properties that enhance their effectiveness and efficiency as problem solving systems.

2.4.4 Bio-inspired MAS

Another attempt that enjoys increased popularity is to draw inspiration from nature where millions of years of evolution have resulted in elegant solutions to overly complex problems, one of the motivations behind the work carried out for the Emergent Distributed Bio-Organisation (EDBO) [145]. This field of study has been referenced with many terms, most popular of which appear to be the bio-inspired or organic computing [134, 146]. Essentially, this approach relies on studying biological systems and concepts that already display autonomous, subsequently modelling solutions applicable to computer systems, ultimately expecting those properties to emerge in the computer systems, too.

Apart from EDBO itself, architectures such as the Organic Grid, where agents attempt to colonise resources they discover in the distributed system much like an array of biological organisms would (humans included), achieve self-organisation and hence a satisfying degree of autonomy [134]. Notwithstanding the fact that organic computing can lead to achieving autonomy in a distributed system, the emergent properties may not always be the ones sought after, or even remotely desirable. This could eventually lead to unwanted behaviour that adversely affects the system, therefore extreme care should be taken when modelling such solutions and extensive evaluation of the final system is of paramount importance [146]. Emergence in bio-inspired solutions will be elaborated through the exploration of EDBO at a later stage in this review.

2.4.5 Distributed Artificial Intelligence

Distributed Artificial Intelligence (DAI) is a subfield of AI that focuses on solving complex problems by distributing computation and intelligence across multiple processing nodes or agents [147]. These systems emphasise parallelism, robustness, and scalability by leveraging multiple computational entities working together to achieve a common goal, much like the MAS discussed above [148]. DAI capitalises on distributed computing infrastructure, including cloud, edge, and decentralised networks, to enhance scalability, efficiency, and fault tolerance in such systems, which are widely desirable traits for applications such as multi-robot systems, intelligent transportation, and large-scale data analysis [147, 149].

By virtue of its nature, the strengths of DAI lie in its scalability, robustness, and efficiency. By distributing workloads across multiple agents or systems, DAI can handle large-scale problems that are computationally infeasible for a single AI system [149]. It enhances fault tolerance by reducing single points of failure, ensuring continuity even if individual agents fail. Moreover, it enables parallel processing, leading to faster and more efficient decision-making. DAI is particularly beneficial in dynamic and real-time environments, such as financial markets, autonomous vehicle networks, and cybersecurity, where decentralised intelligence improves responsiveness and adaptability.

MAS improve upon traditional distributed AI by incorporating autonomous agents that possess individual decision-making capabilities [148]. Unlike DAI, which focuses on distributing computation [149], MAS emphasises agent interaction and adaptability. MAS are particularly advantageous in environments requiring decentralised control, emergent behaviours, and complex inter-agent cooperation [145, 149, 148]. Examples include swarm robotics, automated trading systems, and collaborative AI applications, where the ability of agents to act independently and coordinate dynamically provides superior performance.

The superiority of MAS over traditional DAI arises from their flexibility, robustness, and ability to handle uncertainty [150, 151]. In DAI, coordination mechanisms tend to be rigid, often requiring predefined protocols for information sharing and decision-making [147]. Conversely, with the extensive review of MAS so far, it allows for adaptive and dynamic interactions, making them arguably better suited for real-time and unpredictable environments. Additionally, MAS can integrate learning mechanisms, enabling agents to evolve their strategies over time, which is particularly beneficial in domains like smart grids, autonomous transportation, and distributed healthcare systems, but even more so for the design and goals of this study.

2.4.6 Microservices in distributed MAS

Microservices play a crucial role in enhancing multi-agent distributed systems by providing a modular and scalable architecture where different agents can communicate seamlessly [152]. In a MAS, various independent agents interact to achieve common objectives, often requiring coordination and data sharing. Microservices break down complex applications into smaller, independent services that can be developed, deployed, and scaled separately. This enables each service to be responsible for a very specific functionality, and it communicates with other services using lightweight protocols like HTTP or messaging queues [152]. Consequently, this modular approach ensures that MAS can efficiently distribute workloads, making it easier to manage, update, and scale individual components without disrupting the entire system, which is ideal for autonomous distributed system applications.

Another key advantage of microservices in distributed MAS is their ability to improve flexibility and resilience of the system. Since agents in such systems often have distinct responsibilities, microservices allow for independent service updates, choice of assorted implementations of microservices at runtime, as well as optimisations without affecting the entire ecosystem or requiring a restart [153]. For example, if an agent responsible for data processing needs an algorithm improvement, only its corresponding microservice needs an update, reducing downtime and minimising the risk of system-wide failures. Similarly, if the agent needs to perform differently at runtime when asked to do a specific thing, it can utilise a different microservice implementation for its function without system disruptions. Additionally, microservices enable fault isolation, that is if one service fails, it does not necessarily bring down the entire system, improving overall system reliability and uptime [153, 154].

Microservices also enhance interoperability and communication among agents within a distributed system [155, 152]. Given the fact that each microservice is independent and exposes Application Programming Interfaces (APIs), different intelligent agents can interact without needing to understand the underlying complexities of each other's implementations. This abstraction allows developers to integrate heterogeneous systems and technologies, fostering adaptability and easier collaboration in multi-agent environments [155]. Furthermore, the use of cloud-native microservices enables horizontal scaling, ensuring that as the number of agents increases, the system can efficiently allocate resources and maintain performance [153]. Consequently, by leveraging microservices, distributed MAS can be infused with the necessary qualities that complex real-world applications required, such as the one that is needed for the CASA tracking problem being addressed here.

2.4.7 Proposed architecture for the EDS framework

Distributed system architectures are encountered consistently in our everyday lives, powering our social media apps, e-commerce sites, and video streaming among a wealth of other services, making our digital experiences seamless and efficient. The most higher-level categories of abstraction for these systems can be described by the following three terms: client-server, peer-to-peer (P2P), and microservices [156]. The traditional client-server model has become costly with regards to scalability in distributed applications, due to the need for adding hardware to support parallelism and increasing workload demands [156, 116]. While the problem at hand might not have such needs, it resembles too closely the typical centralised architectures, raising concerns with regards to the ease of introducing emergence on an architectural level, as well as node discoverability and availability.

On the other hand, P2P networks (as EDBO is modelled after) are excellent for such cases, where the nodes can perform several different roles, resources can be utilised and distributed efficiently, and even if a node fails the system can keep functioning; the latter forming a highly desirable trait for the proposed system [157, 158]. Each peer, however, needs to possess all functionalities and smaller subsystems, even though they may not utilise them eventually, introducing efficiency and operational concerns for the component in question. This where the microservices shine, as they handle the problem of functionality responsibility in the best possible manner and, by extent, become easier to scale, maintain, develop, and keep independent; a failure in one microservice does not necessarily dictate a failure of the component [159, 160].

As a result of utilising the microservice approach proposed earlier, the desired system could employ a hard separation of concerns and responsibilities, as per the object-oriented SOLID principles [161], where each node can decide what role they need to assume in a scenario to solve it more efficiently. Consider the separation of listening for sources and gathering information about the environment, for example. A P2P node as a tracking robot should possess both, but this increases the operational costs (e.g. a database for saving data) and with a failure the data can be lost to the rest. A client-server system would keep that information on a server and reduce the costs for the client robots but that introduces another point of failure, more catastrophic for the system than the previous one. But if a data microservice can be hosted in only a few nodes (e.g. on one robot and a mobile phone) and the rest can keep it idle until needed to enable it, overall system performance and cost is reduced.

To instrument microservices several technologies exist, but can effortlessly be narrowed down when considering distributed systems. The key differentiating factor here as opposed to other more specialised approaches, such as streaming services or close-range communicating devices, is the availability of the solution of choice for an array of overlay networks (e.g., Internet, WiFi LAN) to be used for communication. The REpresentational State Transfer (REST) used as a means for realising APIs that operate over HTTP, which the sheer majority of modern smart devices support through networking [162], is the ideal candidate for microservice interaction and state-of-the-art distributed systems [163]. The modern .NET implementations have introduced in the last few years the concept of a Minimal REST API, an approach that manages to serve highly efficient communication with the least required memory and computational power [164].

The other strong contender for implementing the most crucial part of the distributed system (i.e. communication through microservices) is gRPG, Google's Remote Procedural Call (RPC) solution [165]. Indeed, this approach offer unparalleled performance due to memory handling and binary serialisation techniques for contract-oriented microservices and shine in low-latency cases [166]. The contract concept enforces a more static than versatile approach to handling services, whereas in REST the contracts can gradually evolve as needed [165]. Naturally, this comes to contrast with the flexibility desired of intelligent agents and evolving systems, while at the same time passes on the ubiquity that REST offers with its implementation-agnostic approach for heterogeneous communicating devices.

The fact that .NET also supports deploying such APIs on mobile devices, on top of all the major operating systems including lightweight Linux-based distributions, is what nominates microservices as minimal API implementations ideal for the task at hand. The .NET ecosystem also provides several other efficient and modern means to problems that arise in constructing distributed systems, such as mechanisms for node discovery, message passing, lightweight data storage, but most importantly the hosted workers [164]. These workers are an abstraction for the traditional agents that utilise low-level language optimisations in .NET systems to operate with minimal memory and power costs in resource-constrained systems – an excellent candidate for developing the intelligent, bio-inspired agents. Accordingly, a .NET minimal API to describe the functions of its constituents and hosted workers operating as bio-inspired agents will provide a modern solution that can be run on an assortment of heterogeneous devices trying to form an EDS.

2.5 Intelligent Agents

The core elements of autonomy in distributed systems, as well as the catalytic factor in facilitating functionality in that system that can solve the problems they were designed for, are the intelligent agents. This section describes these agents, starting from the older traditional architectures and reaching to the more advanced modern architectures, while at the same time discussing the benefits and detriments of each. This serves the purpose of reaching a conclusion for which strategies should be used at each point of this study and why.

2.5.1 Traditional agent architectures

Intelligent agents are systems that perceive their environment, process information, and take actions to achieve specific goals. These agents are widely used in AI, robotics, and MAS. To differentiate between the structure of intelligent agents, behaviour, and decision-making capabilities, one need only focus on the defining factor: their architecture. This section explores the fundamental and advanced architectures of intelligent agents, including their characteristics, applications, and evolution over time. A The typical classification scheme that can be used to present them uses the three high-level categories of *classic*, *hybrid*, and *advanced* architectures [167].

Classic architectures The oldest and most widely used intelligent agent architecture has been titled *Reactive* [167]. These agents have also been synonymous to *reflex* agents, due to the way they operate they simply use a condition-action rule-set to respond to environmental stimuli as by reflex. A more complex version has been proposed by Brooks, which has been highly utilised in robotics in the past, where hierarchical layers are introduced with the lower levels handling basic reflex actions, whereas the higher levels add any necessary complexity [168]. Consequently, these agents tend to be very robust in dynamic environments, due to possessing fast response times, albeit severely handicapped by the provided rule-sets and incapable of learning.

To overcome such issues in older architectures, the *Deliberative* agent paradigm was conceived. The simple idea behind these agents is to incorporate reasoning and planning to decide on the best course of action based on internal representations of the world [169]. These architectures either use logical representations and inference mechanisms, or keeping an internal world model to plan and execute actions, thereby gaining the capacity to have more complex decision-making especially under unforeseen circumstances, which unfortunately comes at the expense of computational cost that may also translate into slower response time in real-time application [169]. Given that the computational costs in modern day systems are not as significant, this is an ideal approach to tackling issues in dynamic environments as the ones this study will face when learning is not required.

Hybrid architectures The hybrid architectures evolved as a natural combination of both *Reactive* and *Deliberative*, to balance the efficiency of the former with the capacity for intelligence of the latter. The first family of such is that of the *Three-layered* architecture (best described in [170]), which is comprised of: (a) the reactive layer, (b) the deliberative layer, and (c) the executive or sequencing layer. Layer **a** handles immediate response to the environment, layer **b** plans actions using the internal model, and layer **c** coordinates the two layers by translating the abstract plans of layer **b** to immediate actions and ensuring long-term agent goals are respected. Consequently, it keeps the best of both worlds from the two classic architectures it combines, still keeping the main problems of inability to learn, as well as the design complexity for layer **c** that can have problems scaling [170].

The other prominent hybrid architecture is known as *Belief-Desire-Intention*, or simply BDI. The BDI model is distinct from the previous approaches in that it attempts to follow a cognitive path towards designing intelligent agents closer to human reasoning [171], although it still retains some of the same concepts. The agents *Beliefs* is reminiscent of the environment perceived state as encountered in the other architectures, with the *Desires* describing the long-term goals of the agent (e.g. to successfully track a target), and lastly *Intentions* encompassing the commitment to specific actions as a differentiating factor form the previous ones [171]. The greatest strength of this model lies in the effective handling of goal-driven behaviour, while at the same time offering adaptability under uncertainty, nonetheless the computational cost it introduces in tandem with the complexity towards designing for dynamic environments [172]. Both disadvantages can constitute severe handicaps for the study at hand.

2.5.2 Proposed architectures for the bio-inspired strategies

The bio-inspired strategies are simple by virtue of their design, which in turn is based on simple observations from the biological studies relating to treefrogs. The traditional strategies are hence adequate for modelling the tracking behaviours in the form of strategies during the first attempts at solving the problem. The three steps in this process have been presented as: (a) purely bio-inspired behaviour, (b) advanced behaviour based on the findings from the earlier experiments, and (c) a truly adaptive behaviour that combines all the findings and learning capabilities. Strategy **c** cannot be addressed by the traditional strategies given their lack of learning capabilities, so it will have to rely on advanced techniques that will be discussed later.

For strategy **a** it is imperative that the simple observations provided by the biological strategies are converted into intelligent agent strategies as accurately as possible. The rules describing these behaviours are elementary and straight-forward, as they are not accompanied by information related to cognitive functionality and decision-making, given the natural inability to know the mind of treefrog. This is a boon for the design of such strategies since the simple rules can easily be covered by the most elementary and performant, as well as easy to implement, *reactive* architecture. The rules can be designed as such: type of frog dictates a chance for listening vs. a chance for moving, and the agent then takes the action. This keeps the strategy as close to biology as possible, while at the same time allowing to establish a much-needed baseline for the performance of the advanced strategies to follow them.

On the other hand, strategy **b** will have to introduce to the functionality of the agents beyond what is covered by the bio-inspired strategies. Moreover, even if this strategy will still not rely on learning, it does require a new means of applying the lessons learned from the experiments with the bio-inspired strategies. This can be designed as a new layer that provides such reasoning capabilities for the agents, which could allude to emulation of cognition on the intelligent nodes of the system. Both prominent hybrid strategies do allow for this, nonetheless the implementation complexity they introduce is something that should be avoided. Especially so given the fact that several advantages they bring (e.g., the beliefs and desires from *BDI* or sequencing from *Three-layered*) would require a more intricate breakdown of the cognition layer that will be required, which might not be possible due to the yet unknown experiment results. Naturally, the *deliberative* model with its simple decision-making based on a recorded state of the world (as perceived by the agent alone) can provide what is needed for this case.

2.5.3 Advanced agent architectures

The agents that would be capable of an adaptive strategy, which will be able to address changes to the environment on the fly, need to be a step up in complexity from the traditional ones to be able to carry out their task. Naturally, this leads to needs for a consideration of the proper AI techniques to use, the ones infused with learning capability that the traditional ones lack. RL stands out for its ability to learn from interactions, handle delayed rewards, and make sequential decisions, constituting a suitable AI approach for dynamic scenarios where immediate actions impact future outcomes [173, 174]. As a matter of fact, this is a perfect description of the current research goals, but the other two predominant Supervised and Unsupervised Learning (SL – UL) approaches need to be considered before a choice is made.

The SL machine learning paradigm aspired to teach the prospective intelligent system to learn a mapping from input data to desired outputs. This is achieved through what is labelled as the training dataset, a pair of input data with expect outputs, which is continuously supplied to the system with varying inputs and outputs, so that the system can them attempt through the implemented logic to derive a mapping function. To achieve that the algorithm needs to be able to generalise from the unknown inputs to a reasonable output. The iteratively developed and enhanced function receives constant and explicit feedback during training sessions. Naturally, SL is typically applied for scenarios where mappings are easier to derive, such as speech and handwriting recognition, language translation, and image classification [175].

On the other hand, UL refers to fundamental AI techniques that attempt to understand the provided data without relying on specifying the expected outcomes. Essentially, it revolves around discovering distinct patterns, hidden relationships, or larger structures in non-categorised input data. Typical methods for achieving these difficult tasks are the clustering of similar data points together based on certain features, identifying associations or co-occurrences among items, and attempting to reduce the number of discovered features whilst preserving information deemed crucial for the data. The core idea, as per its literal definition, is to not intervene in the learning process and leave the system to operate in an autonomous manner. The typical use cases for a UL system include providing recommendations, detecting anomalies in provided data, as well as improved feature extraction [175].

RL is a machine learning paradigm with a focus on enabling agents to take actions in a dynamic environment through sequential decisions that will eventually lead to maximising the cumulative rewards, however these are modelled in the system. No training data, whether inputs or expected outputs, are provided in this pure case of RL. Instead, the AI interacts with its environment and learns how to choose the next action based on rewards or penalties, which constitute the training feedback. To realise its purpose, a RL agent employs the concepts of exploration and exploitation via a balancing act. To elaborate, the agents find themselves in dynamic, unknown environments, with a risk of performing unrewarding actions when they explore, or at the cost of missing better alternatives when exploiting familiar approaches. Abundant applications can be found in robotics, as well as game playing, or even recommendation systems accepting user feedback for their accuracy [174].

The key advantage of RL over SL and UL is that it promotes learning through interaction with a dynamic environment, while these resort to work with static datasets. The prominent disadvantage can be that RL receives delayed rewards, as the agent must live for a time in the environment and attempt to adapt to it, whereas SL and UL do not for cases where training happens at problem-solving runtime [175]. SL is extremely handicapped with regards to the mobile audio source tracking problem in this specific dynamic environment. The static datasets for such a scenario are immense and the environment is ever-changing, considering the very nature of mobility and randomised obstacles along the way.

Meanwhile, typical UL such as with neural networks suffers from similar problems of requiring immense training datasets, albeit halved as compared to SL (i.e. only input), yet the pattern discovery such systems are lauded for can be beneficial only in a theoretical basis for the problem at hand – in practice, the agent will need to re-evaluate its decisions every step of the way towards the target with a high processing cost, whilst the act of developing such training datasets could be an entirely different study of its own [176, 175]. And this is where RL shines, in that it is designed to work in making sequential decisions, step-by-step, mostly relying on exploration at the start and resorting to exploitation at the end. As a matter of course, this establishes it as the prime AI technique for the complex adaptive strategy of the targeted problem.

2.5.4 Reinforcement learning

The goal of this thesis is to infuse intelligence in the tracking nodes to foster adaptivity in the tracking behaviour, after the preliminary experiments with the bio-inspired strategies. RL is an interdisciplinary area of machine learning and optimal control that is concerned with the capacity of an intelligent agent to take actions within a dynamic environment with the goal of maximising their cumulative rewards [177]. Optimal control theory focuses on contriving an algorithm through variables that the system can manipulate over the course of time to minimise cost to satisfy design objects [178]. The cumulative reward is the concept that can be used to minimise the cost function [177]. The interdisciplinary nature and focus of RL align with the goals of the expected system that can solve the problem.

In RL the agent life cycle can be described as follows: the agent interacts with the environment, receives a reward – or possibly a penalty – for the action that could come at a delayed time, and receives an updated state. Unlike SL, RL does not require input-output pair training data, rather it operates with a *trial-and-error* strategy, eventually learning by interacting with the environment. This introduces the ideas that have been mentioned previously in the study, the *Exploration* (trying new actions) vs. the *Exploitation* (using known actions) to maximise long-term rewards. The environment is typically modelled as a Markov Decision Process (MDP), which defines the transition probabilities between states based on chosen actions [177, 174].

MDP is defined by the following 4-tuple [177]:

$$(S, A, P_a, R_a)$$

S	The set of States.
A	The set of Actions.
$P_a(s, s')$	The probability for action a to lead to state s' from state s.
$R_a(s, s')$	The expected or immediate reward for action a when transitioning from state s to state s'.

Cardinal concepts of RL as related to the above is the state space, the action space, whether these two are continuous or discrete, and the policy for control. Discrete states characterise finite or countable states (e.g. tic-tac-toe game), whereas continuous states are uncountable and often represented by continuous result functions of multiple parameters (e.g. robotic arm control) [177, 173]. Discrete actions refer to finite, distinct actions (e.g. placing either `X` or `O` on the tic-tac-toe board), whilst a continuous action set define continuous range of possible actions (e.g. applying force to the robotic arm joint). In discrete spaces exploration strategies tend to be straightforward, compared to the challenge of infinite possible actions that typically require neural networks to reach the best action [179]. Policy defines the decision-making strategy the agent employs in RL, providing the mapping function between current state and action. Policies can be either deterministic (e.g. move every time) or stochastic (e.g. stay still 3 out of 10 times, move the rest) [173]. In policy-based RL the agent learns a good policy through the iterative process of exploring an action, evaluating its impact on rewards, and updating the policy accordingly [173].

A highly attractive variant of RL is the Inverse Reinforcement Learning (IRL) [180]. This machine learning approach shifts the focus to inferring the underlying rewarding function from observed behaviour. IRL can find applications in real-world scenarios where there is no access to explicit reward functions by means of observing the results of other more successful agents or policies [180, 181]. Consequently, given expert demonstrations the IRL mechanism devises a reward faction that can assist the agent in mimicking the expert agents behaviour and performing better [181]. The most detrimental challenges with IRL are: it is an under-defined problem in the sense that it lacks a unique solution and multiple reward functions could elicit the same results, and they require a large number of expert demonstrations which can be infeasible to obtain [180].

Moreover, there are RL techniques categorised as off-policy [173, 179], which allows the agent to learn from historical data collected by a different policy than the currently active one. In such cases, there is a target policy that needs to be improved by using a different, behaviour policy that interacts with the environment. This approach allows for historical data re-usability and provides sample efficiency due to learning from diverse data without additional exploration. A more noteworthy trait, however, is that the behaviour policy can explore widely, a key requirement in dynamic tracking scenarios, whereas the target policy can exploit the best actions [177, 179].

In closing, there is also an alternative to policy-based RL: value-based RL [173, 177], where the focus is on estimating the value of different states or state-action pairs, essentially developing a value function as a policy. There are two value functions to consider: (a) $V(s)$ which is concerned with state and what rewards can be gained from being in it, and (b) $A(s, a)$ which focuses on the right action from a specific state. Relying primarily on Bellman equations, they aspire to update value estimates based on future rewards. For the balancing act between exploration and exploitation, strategies like ε-greedy (pronounced epsilon-greedy) offer optimal results. True to its name, it chooses a greedy action to get the most immediate reward by exploiting the current action-value estimates he agent knows, which generally leads to favouring good known actions by being only occasionally exploratory [179]. It is evident that value-based strategies shine when the states and actions can be discrete, aligning perfectly with the *stop-perceive-act* paradigm the in robotics. For a high-risk undertaking such as the tracking of mobile targets with energy concerns, this could be a candidate for an optimal solution.

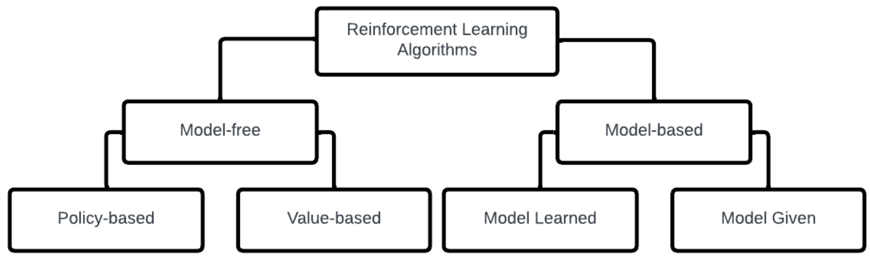

Figure 2.3: A high-level taxonomy for reinforcement learning algorithms, categorised based on the predominant algorithm properties.

2.5.5 Reinforcement learning algorithms

Figure 2.3 illustrates a possible high-level taxonomy of RL algorithms, showcasing the first choice being between a model-based and model-free algorithm. The former attempts to choose the best policy based on the model of the environment (either provided or contrived), whereas the latter chooses the optimal policy based on a trial-and-error approach [182]. In model-based algorithms the agent can decide what action to choose by knowing what is expected to happen based on the known model, a property that does provide higher sample efficiency than trial-and-error, albeit at the cost of focusing occasionally on irrelevant details. This can have a detrimental effect on an evolving environment and the model-free approach will ignore such details by default. Another strength of the model-free approach is, naturally, the ease of implementation and use given the lack of a binding model to factor into the assorted functions [182, 177].

A second level of taxonomy visited earlier, could constitute the policy-based as opposed to the value-based approach. Policy-based RL directly optimises the agent's policy, which is a mapping from states to actions, without relying on a specific value function. Instead of estimating action values like in value-based methods, policy-based methods learn and improve a parametrised policy using assorted techniques to maximise expected rewards [177]. One fundamental difference between these two categories that can have a high impact on providing a solution to the problem at hand, is the higher variance and slower adaptation of policy-based vs. value-based [182]. The value-based methods try to pick the action which maximises the action-state value function which will improve the policy in the direction to the best policy – and by extend in a faster manner and with lower variance, while policy-based methods attempt a little step and smoothly update in that direction estimated to be more stable but in the same time less efficient and sometimes leads to higher variance. This property also drives policy-based approaches to converge to a local optimum instead of a global, which is desired for the scenario at hand in the long term. The concept of convergence in RL refers to achieving the optimal policy after an arbitrarily long period of time [183].

Finally, there are off-policy and on-policy algorithms: the former uses a different policy for exploring and uses the results to improve the different exploiting policy, the core balancing act in RL, whereas the latter attempts to improve a single policy. Beyond the freedom given to exploration in off-policy RL highlighted earlier, there are other desirable traits to capitalise on. In particular, off-policy tends to learn a better policy eventually, as opposed to a safer policy that its counterpart attempts to achieve, albeit at the cost of convergence time [183, 182]. Off-policy also is great at avoiding getting trapped in local minima in its function thanks to the wider exploration behaviour, too [182]. In conclusion, these traits crucial for the specific scenario of energy-efficient tracking of sources in a dynamic acoustic scene argue for an algorithm that is model-free, value-based, and off-policy, which describes the dominant traits of *Q-learning*.

2.5.6 Proposed adaptive strategy solution: Q-learning, ε-greedy

Q-learning [184] is an algorithm that exhibits the RL algorithm traits that have been deemed necessary for the current agents, and it is important in how it expertly realises them. The core concept of Q-learning are the Q-values (Q stemming from Quality), which are assigned to each state-action pair for the agent to know value for a specific action in a specific state. These are used to create the Q-function, in essence a look-up table that maps actions to states and holds a value in each cell, referred to as the Q-table. These Q-values usually start at 0 and are assigned a proper value as the agent explores the environment and gets a reward (or a penalty) for the action compared to the new state. In the end, the action-reward iteration until a terminal state is reached keeps updating the table until it reaches the state where it holds the optimal Q-function. One training session, start state to terminal state (i.e. reached a target or out of energy), comprises an episode [184].

to update the Q-values, especially at the beginning, the agent needs to explore first. In Q-learning the agent has a behaviour policy (off-policy) that determines if the agent should explore to update the main policy (Q-function) or exploit it instead. This policy is called ε-greedy, at the centre of which is the variable ε that takes values up to 1.0, and defines through a random value between $0-1$ if the target explores (value between $0-\varepsilon$) or exploits (value between $1-\varepsilon$). Consequently, training starts with an ε at max so that the agent can explore and start creating values, eventually lowering this value in consequent episodes to start exploiting and finding better Q-values. This is how the agent can ideally learn by interacting with the environment repeatedly [184].

One important note is that Q-learning belongs to another category of RL algorithms, that of Temporal Difference (TD) [182]: get a reward immediately after an action instead of after concluding the episode. This aligns with the problem at hand where every action has an associated energy penalty, which can readily factor in the reward assignment function by teaching the agent eventually to minimise the energy cost. The last concept of importance to RL and Q-learning is the updating of the Q-value, which is calculated using an adapted form of the Bellman Equation [184]:

$$Q(S_t, A_t) = (1 - \alpha)\, Q(S_t, A_t) + \alpha\, (R_{t+1} + \gamma \cdot max_a Q(S_{t+1}, A))$$

$Q(S_t, A_t)$	The new/former Q-value (left/right of the equation respectively).
S	Represents a state.
A	Represents an action.
t	The current time step ($t + 1$ being the next one).
R_{t+1}	The reward for taking an action for the next time step.
α	The learning rate.
γ	The discount factor.

The state, actions, and rewards are familiar concepts to this point, with two key factors being introduced that define the success or failure of the optimal policy training can be identified within that equation: α, the learning rate of the agent, and γ, the discount rate on future rewards. The new Q-value for taking an action thus considers all these parameters, and the learning and discount rates naturally affect the outcome heavily. Accounting also for the exploration rate ϵ, which affects overall performance of the strategy, the list of factors that can affect the training of a policy are expanded by one.

Naturally, the three variables that heavily affect training are:

α Determines the learning rate of the agent – higher rate means more aggressive updates; lower rate means slower updates. The learning rate has a high impact on convergence for the optimal policy: a higher rate demonstrates earlier convergence with a risk for sub-optimal values and instability, a lower rate reaches convergence more slowly but reduces the risk of overshooting in values. This variable takes values between $0 - 1.0$, with 0 indicating that the agent learns nothing from new actions and 1.0 meaning that it entirely ignores prior knowledge.

γ Determines whether the agent relies more on short-term or long-term solutions. Consequently, with a lower rate the agent learns to look for the best possible immediate rewards, whereas a higher rate teaches the agent to accept a punishment in immediate rewards so that it may get higher rewards later. Environment plays a key role in whether short- or long-term rewards are preferable. This variable takes values between $0 - 1.0$, with 0 instructing the agent to totally ignore future rewards and 1.0 to really focus on obtaining high rewards, although at the cost of getting stuck in local optima.

ε Determines the exploration rate for the agent, as explained above. Plays a crucial role in allowing the agent to escape the local optima that could be introduced by other factors (e.g. high γ value).

The above variables are also referred to as hyperparameters and their tuning process is essential for the success of the training process due to co-dependencies [185, 186]. The balancing act between exploitation and explorations for the success of the strategy needs to be expanded to include balancing of learning rates and rewards. These will be addressed in the section devoted to the on-policy training later. In closing, Q-learning as a machine-learning means of infusing an agent with the capacity to address the mobile sound source tracking problem in an energy-efficient manner can be stated as fitting. It is free of a model, with a design to look for the optimal approach with lower variance and in a faster time, while avoid getting stuck in local optima and being easier to implement – an important aspect given the constraints and focus of this thesis.

2.6 Emergent Systems

The final step in this study is to harness emergence in an autonomous distributed CASA system, where devices attempt to solve the complex, energy-efficient, tracking problem in tandem. This section gives a brief overview of what *emergence* is, how such systems incorporate it and what its effects can be, as well as why it is desirable for this study. The core inspiration behind the proposed system is presented through the lens of its conceptual predecessor system, while the research field of solving problems in a collaborative manner via pertinent behaviours derived from research in social sciences, is also investigated for solutions that can be incorporated into the final solution this research work proposes to foster emergence.

2.6.1 Emergence in complex systems

In large entities, such as complex systems, comprised of or populated with smaller entities in nature, it can be observed that the interactions among these smaller entities allow for the larger entity to possess properties that the smaller entities do not possess [187, 188]. This phenomenon has been called *emergence* and it has been encountered in several disciplines, as well as spanning across them, such as the phenomenon of life, a living working organism studied by biology being based on the interactions of the chemical elements in the body studied by chemistry, or a flock of birds exhibiting behaviour not found in each single bird studied by both biologists and the social sciences [189].

Emergence in general may also be divided into two distinct categories, weak and strong emergence; a division that is based on the capacity of an emergent property in a system to be simulated with computers or not [190, 191]. Weak emergence is focused on the typical concepts of emergence, where interactions at the lower levels produce properties in the high-level system, with the core concept being that these properties, whether desirable or not, have to be determined through the observation or simulation of the system [190]. Strong emergence, on the other hand, is rooted in the idea that there are systems that cannot be simulated, essentially advocating that in these systems attempts to simulate emergent properties can only lead to the breaking down of a system to its parts that possess these properties, rather than parts that interact to produce them [191].

2.6.2 Architectures for emergent systems

In the field of computer science, weak emergence is the key to developing systems with desirable properties. Examples include the engineering of emergence in MAS populated with software agents that are capable of infusing the system with such properties through their basic interactions [192]. Such systems are usually modelled after complex systems found in nature, simulating properties and behaviours such as the swarming of birds, ant colonies, or even human societies and activities (e.g. stock exchange). The goal is to apply these findings to improve efficiency and effectiveness of assorted computer systems. Regarding the autonomous systems domain in particular, emergence has been explored with regards to the capability of providing microscopic properties to the constituents of the system, consequently anticipating in the macroscopic level self-organisation capabilities for the system when the constituents interact, or as the system grows.

Emergence is something that is unpredictable, something that can have both negative and positive repercussions, and from the definition of emergence it can only be determined when the system has been developed and in operation, through either simulation or observation. This begs the question of how it is possible to engineer emergence in computer systems. Some of the answers lie within the existing research and results of past endeavours, such as the computational, bio-inspired solutions to problems [188]. The optimal approach, nonetheless, is to begin with the design of a high level, abstract model that is no manner influenced by the expected implementation as a computer system, rather it is infused with creativity in design that may not be part of a standardised solution [193, 188].

In the more recent and advanced realm of artificial intelligence, generative models like GANs – much like the recent advancements in CASA discussed earlier – and reinforcement learning algorithms are also fostering emergent behaviour in complex systems, whether distributed in nature or not [194]. GANs generate realistic data samples by pitting two neural networks against each other in a game-theoretic framework, leading to the emergence of novel and diverse outputs [194, 195]. Similarly, reinforcement learning agents learn to make sequential decisions by interacting with an environment, often exhibiting emergent strategies and behaviours beyond what was explicitly programmed [177, 174].

2.6.3 Emergent Distributed Bio-Organisation

The Emergent Distributed Bio-Organisation (EDBO) is a model based on work focusing on the exploitation of emergent phenomena that can be developed within distributed systems. Initial work on understanding the interaction between microscopic and macroscopic links among properties of the agents and system respectively, has been the driving force behind the development of a disciplined framework for engineering emergence [145]. EDBO follows the best practice of beginning with a creative, high-level, abstract model with an aim to engineer emergent properties at the macroscopic level, with further work investigating its implementation feasibility and capabilities as distributed system.

EDBO is based on the concepts of weak emergence for software systems, focusing on properties and interactions at the microscopic level to enable the development of potentially desirable properties at the global, or macroscopic, level. Most bio-inspired approaches tackle specific problems with limited scope, whereas EDBO aspires to provide an expanded, a holistic, approach to tackling multiple operational issues that distributed systems suffer from. EDBO has been instrumental towards the inspiration for this thesis and its first steps, developed as a distributed system first [196] and then transformed into a middleware for Internet of Things (IoT) applications [197].

As an abstract model, the scope of EDBO is limited to the topmost layer of the design of autonomous distributed systems. The formulation of a basic, unstructured distributed system model has been devised so that it is infused with properties that allow for a high level of availability, scalability, and robustness, under assorted operational conditions. By introducing biologically-inspired properties in the software agents at the core of the MAS the model describes, EDBO utilises techniques of the agent-based paradigm to prepare the ground for desirable phenomena to emerge in the system. These agents represent the network nodes and are called *BioBots* and reside in what is referred to as the *BioSpace*.

BioBot behaviour is based on several bio-inspired mechanisms that guide its decision-making process, which governs life-cycle functions such as energy management, migration, reproduction, replication, birth and death. These properties have been introduced in the model in an abstract manner and could therefore be transferred to various different functionalities in implementation scenarios (e.g., robots equipped with sensors migrating to better acoustic positions, agents reproducing to serve a higher volume of concurrent user requests). The life-cycle heuristics assess the state of the BioBot with regards to energy levels and make decisions regarding which functionalities are available and how many relationships can be supported.

The model currently proposes two different energy types:

Discovery energy The discovery energy is awarded to the BioBot that succeed in completing the discovery process.

Service energy Service energy is allocated to each BioBot interacting with each other that managed to succeed in the discovery process.

This is a prime example of emergence: introduction of the above microscopic properties generating macroscopic emergent behaviours through agent interactions. In the natural world, biological organisms aspire to maximise the energy gain and minimise the energy loss as a vital principle for survival. As an example, BioBots that keep losing energy will end up dying, however that translates into implementation specifics (e.g., node removed from network, sensor temporarily powering down to conserve energy), while successful energy management is rewarded with the ability to expand relationships and reproduce.

This adaptation through natural selection is the product of fitness-based evolution, primarily through energy management, which is relevant to the CASA problem this thesis attempts to solve via energy-efficient tracking. By consequence, introducing such microscopic properties to the CASA framework to be developed, via the modelling of bio-inspired strategies and their evolution, desired behaviours that offer an optimised solution to the problem at hand can emerge. This approach to problem-solving and optimisation has been a key influence in the design of this study: treefrog-inspired strategies evolved into advanced approaches and using meaningful interactions among the agents to foster better problem solving, which means collaboration. Infusing collaboration is what should be discussed next through the lens of social sciences.

2.6.4 CPS in social sciences

Collaborative Problem Solving (CPS) in social studies involves a group of individuals working together to identify, analyse, and resolve complex social or mental issue, even real physical challenges [198, 199]. In this approach, participants utilise their entire arsenal, pooling their knowledge, skills, and perspectives to attempt to address problems that may be too intricate or multifaceted for any single person to solve alone – the community can achieve the goal instead, is the focus of this field. CPS in social studies emphasises collaboration, critical thinking, communication, and empathy as part of the essential skills for navigating these assorted complex issues as effectively as possible [198, 199].

In a CPS setting within social studies, participants engage in activities such as group discussions, debates, case studies, and simulations to explore real-world problems [200, 198]. These problems could range from issues related to politics, economics, culture, education, or social justice; however they can also relate to physical activities such as solving a problem harnessing the powers of each individual in the community towards the best possible outcome. Through collaboration, participants draw on their diverse backgrounds, experiences, and possibly physical skills where they excel at so that they can get into the root causes of these problems and develop potentially efficient and effective solutions [200, 198].

Additionally, CPS in social studies encourages active engagement with different perspectives and encourages constructive dialogue among participants [201], albeit with the caveat of balancing what has to be communicated and how to achieve the desired degree of efficacy in the CPS endeavour [202]. By considering various viewpoints and evaluating evidence, individuals can develop a deeper understanding of complex social phenomena and the interconnectedness of societal structures [203, 204], a core aspect of emergence found in CPS [203]. This collaborative approach fosters critical thinking skills and promotes civic engagement by empowering individuals to take informed action to address social and physical challenges within their communities and beyond.

Overall, CPS in social studies serves as a valuable tool for promoting social awareness, empathy, and collective problem-solving skills essential for navigating an increasingly complex and interconnected world [198, 203]. Consequently, presenting optimal solutions via CPS essentially revolves around finding the proper way of organising such communities of individuals, by providing models that assist the undertaking to learn, communicate, and apply their skills efficiently [198, 199, 203, 205]. In social sciences, this field is vast and involves an immense array of microscopic behavioural and intelligence-related properties that not all are applicable to intelligent agents that are the focus in this thesis, which dictates the need to look for traits that can be applied to the case at hand and utilised in software systems.

2.6.5 CPS for intelligent agents

CPS employing intelligent agents is a methodology that relies solely on Artificial Intelligence (AI) instead of humans to address the complex problems a software system could pitted against. In this framework, intelligent agents interact with each other, sharing information, analysing data, and generating solutions autonomously, nonetheless with a view towards contributing to the overall solution with their individual part [206, 207]. These agents encompass a spectrum of AI technologies, from basic algorithms to sophisticated systems capable of learning and adapting [200, 207, 206]. Naturally, the objective is to harness the collective computational power and problem-solving capabilities of these agents to attain the more efficient and effective problem resolution.

An essential aspect of CPS using intelligent agents is the integration of diverse AI techniques and methodologies, obviously dependent on the eventual real-world application of the system, which ranges from elementary RL to more advanced neural network implementations and – the more recently and rapidly advanced – conversational type agents [208]. Consequently, having reviewed such fields earlier in this thesis, such agents leverage various approaches – machine learning, optimisation algorithms, and natural language processing to name a few – to collaborate in problem-solving tasks. Going a step further, by combining different AI methodologies, they can tackle multifaceted problems that may be beyond the capacity of individual agents [208, 204, 200]. This collaborative approach thus enables intelligent agents to explore a broader solution space, leading to more robust and innovative outcomes that can be exploited.

Moreover, CPS using intelligent agents facilitates decentralised problem-solving across distributed networks [209, 207]. Through communication protocols and data exchange mechanisms, intelligent agents can collaborate seamlessly across different locations, such as by employing overlay networks locally or even the Internet in global scale, and assorted computational environments: smart devices, smartphones, robots, servers, satellites, etc. This decentralised model enables scalable and parallelised problem-solving, as highlighted repeatedly when discussing the autonomous distributed systems, allowing intelligent agents to address large-scale and dynamic challenges efficiently.

Ultimately, CPS using intelligent agents represents a paradigm shift in the human-typical CPS methodologies, harnessing the collective intelligence of AI systems to tackle complex problems, autonomously but in collaboration, and effectively. Once more, the problem that intelligent agents and AI techniques must face when it comes to adapting the social – or biological as evidenced in previous parts of this work – science findings, techniques, and models, is that of how to properly translate these properties to something that the software and hardware at hand can utilise and capitalise on. Feasibility of such implementations is a core issue, as is the meaningful selection of what is communicated [201, 210]. These will be explored next in the context of emergence.

2.6.6 Socio-cognitive traits for EDS

Decentralised systems naturally allude to the Emergent Distributed Systems (EDS). The developed EDS framework is put to the task of solving the energy-efficient tracking of mobile audio sources in dynamic acoustic scenes through CPS – the culmination of this thesis. The aim is to touch on the surface of emergence by performing preliminary studies in facilitating CPS through unique interactions. As discussed earlier when exploring the field of emergence and through EDBO, these microscopic properties that can be introduced to the agents may eventually produce desirable effects at the macroscopic level that contribute to a more efficient and effective solution to the problem. Crucial element in the success of such undertakings is hence the investigation of which micro-properties to introduce to the agents so that they could eventually achieve their goals better.

The field of emergence is vast and the applications, research directions, as well as optimisation opportunities abound. Careful design and planning are required to determine which micro-properties could have macro-effects beneficial and not unintended for the problem being addressed. This requires meticulous investigation of work pertinent to CPS that could be adapted to software and intelligent agents in a manner that would assist the problem-solving capabilities of the system. What this endeavour also entails is analysis of opportunities in the software itself, which have been made clear through the design and architecture of the expected system from the start: communication between the nodes of a distributed system. Naturally, the focus on background research narrowed down the research into the field of CPS to communication between the interacting individuals in social sciences, and how it can be incorporated into the architecture of agents.

2.6.7 Proposed social traits

There is wealth of traits and behaviours that factor into CPS performance for human individuals. Prominent qualities for humans that contribute to CPS appear to be the broader categories empathy, communication skills, critical thinking, leadership and fellowship, as well as flexibility and adaptability, all of them to varying degrees depending on context [211, 202, 212, 213, 203]. Trait theory in social sciences groups such qualities under specific traits categories, the highly popularised since the 1980s "Big Five", or OCEAN model from the initials of these give important traits that shape personality and, by consequence, the entire life of a human [214, 215].

The OCEAN traits are:

Openness inventive/curious vs. consistent/cautious

Conscientiousness efficient/organised vs. extravagant/careless

Extraversion outgoing/energetic vs. solitary/reserved

Agreeableness friendly/compassionate vs. critical/judgmental

Neuroticism sensitive/nervous vs. resilient/confident

Naturally, computer science work related entirely with the problem of CPS on its own, as well as its immediate impact, focuses on realising these traits in their EDS or MAS for simulation purposes and studying emergent behaviours in that context [216, 211], albeit not within the context of CASA or related solutions that have eventual repercussions in the physical world as applications that attempt to solve such a problem. Evidently, such mechanisms could be implemented for the case at hand, nevertheless not in a timely manner for its scope and goal, leading to the need for distilling the important lower-level microscopic properties that are common in such characteristics that the agents can effortlessly implement and utilise given the developed EDS. Indeed, intelligent agents tend to capitalise primarily on properly developed communication that fosters cooperation in tandem with problem-solving orientation [217, 218, 219].

These studies and the properties of the traits above highlight the importance of some traits that can seem elementary and microscopic, as desired in emergence studies, such as being *inquisitive* or open to *sharing.* The former can combine *openness* and *agreeableness*, provided that the agent is curious to learn what other targets it could track and willing to consider the information shared by its peers, whereas the latter can combine *extraversion* and *neuroticism*, which can be done through the willingness to share and help others while at the same time trying to be confident in helping others or nervous about own performance.

Consequently, these two traits can easily be integrated to the behaviour of the agents who are aware of what characteristics of a signal they are tracking and its position, as well as through the employment of the communication subsystem implemented. The *high*/*low* values desired for each can thus govern: (a) for *inquisitive* how often they want to learn more and how willing they are to accept the information, and (b) for *sharing* how often the share information about known targets and their tracking confidence.

2.6.8 Proposed cognitive traits

Cognition plays a key role in the capacity for general-problem solving, which does factor into the domain of CPS. Indeed, CPS attempts to leverage the best problem-solving skills of each entity for each sub-task that the complex problem can be broken down into, hence the reason why intelligent agents are expected to perform autonomously first and foremost, and only then considered how they can perform the task optimally in collaboration with other entities. Accordingly, the primary trait that cannot be discounted and absent from the study is the core problem-solving skill that the strategies have been attempting to address up until now: the energy-efficient tracking of targets in the acoustic scene. Indeed, this neatly combines some of the predominant base cognitive traits into one: memory, perception, executive function, and spatial awareness [220].

To this end, thinking on performance capabilities associated with CPS and the *high*/*low* distinction between the traits, the developed tracking strategies can be employed respectively: *adaptive*/*combined*. Critical thinking, which the former strategy has developed much more intricately through Q-learning, is an integral aspect of CPS [212, 211, 216] and the weakest aspect of the latter. This constitutes the *problem-solving* trait one of the most important in the outcome of the EDS and its capacity to address the problem it has been developed for. The intelligent agents are thus expected to possess one of these two strategies that are the more performant and combine it with the other social traits and another cognitive trait towards realising the socio-cognitive AI entities.

One key aspect of cognition beyond the problem-solving skills is the expertise an individual in the CPS community possesses, which is also interacting with critical thinking and contributes greatly towards knowledge creation and exploitation [212, 213, 211, 221]. Part of the expertise can be found in the developed problem-solving skill of the individual, yet a more obscure and interesting part of it can be tied to self-confidence that has an immense impact on decision-making and eventual achievements in any undertaking, whether positive [221, 213, 220] or negative [222, 213], however also it may have even social impact [223, 220].

Naturally, *self-confidence* can comprise the second desired microscopic property of the agent that can affect CPS performance for the agents. This new cognitive trait is chosen among several others that have been discounted, such as attention, creativity, and introspection [220]. This can be attributed to the lack of developed mechanisms to support these implementations, or ease of adopting and training new ones from other pertinent work on the field within the scope of this work. *High*/*low* values for this trait can describe an agent that trusts their own state of the world and problem-solving skills more than what other agents can share with them. This can be facilitated by not attempting communication too often, as well as by not considering information coming from others and relying on own faculties, for *high self-confidence* cases as an example.

2.7 Summary of findings

The biology of hearing delivers a vast space in which to study behaviour patterns and qualities of auditory systems and hence develop aids and tools to tackle specific problems or allow to capitalise on the strengths of existing tools to solve bigger, more complex problems that tend to be overlooked. ASA is a multi-faceted area which intersects various research fields, as well as sub-disciplines of each one. Consequently, novel research in this area can benefit from interdisciplinary work be seeking concrete solutions in each field that can be brought together to solve important practical problems. Indeed, the work should not necessarily be confined to human hearing. Over the years, observation and modelling of animal behaviours has produced optimised solutions – at best for a variety of problems narrow in scope but also contributing to tackling the grander problem of auditory perception.

CASA is the key to exploiting advancements in both ASA and technology to solve problems that could have high impact to our life, such as speech perception and localisation of talkers for monaural people, or even guidance aiding for blind people. Advancements have led work in fields such as neuroscience and probabilistic theories for optimising performance, shying away from mere min-maxing of spectral analysis input. Research is usually confined to solving very specific, small-scale problems in the most efficient and effective manner, which inadvertently results in multiple highly heterogeneous end-systems, when the need arises to compile a more holistic solution to the auditory perception system in general. Mobility of the listener and/or of the audio sources is something that tends to be disregarded in many cases and can provide a wealth of information to work with and exploit within a CASA system. Tracking mobile targets hence constitutes an undertaking that can further advance the field.

Autonomous distributed systems are essential to the realisation of scalable and robust solutions utilising networked, heterogeneous devices. Autonomous operations of such systems can be bolstered with the utilisation of agents at the backend handling all functions, while bio-inspired approaches can offer selective optimisation of sub-routines or new solutions to old problems, much like it does with ASA/CASA. Here the opportunity arises to create a modern, distributed system that can provide such functions through AI, while operating as a networked listening device. Meanwhile, the reliance of such systems on energy for autonomous operation inspires the utilisation of bio-inspired tracking strategies for the intelligent nodes and is something that is lacking in related research.

Emergence is a field that can take roots in most disciplines, bringing out the best of bio-inspired concepts and applications in them. The most basic requirements for implementing emergent systems are bio-inspired properties in their constituent sub-systems, notwithstanding the need for interactions among them. Engineering emergence starts at a conceptual level with a model of a system found in nature, followed by the faithful reproduction of those properties in a microscopic level (e.g. a bio-inspired nodes in an EDS), and finalised with evaluation of the system at runtime. Iterative design, implementation and evaluation is the key to causing the desired macroscopic properties of a system to emerge. The inspiration from EDBO drives the ideas behind the evolved design for AI capable of basic RL and interactions with socio-cognitive properties that attribute to success in CPS scenarios.

Putting it all together, there is an intersection of all those fields where current research has not delved into sufficient depth, yet: solving the CASA tracking problem using bio-inspired approaches that may be bolstered by emergence with the goal of minimising energy costs. What the outcome of this thesis could provide is a distributed system powered by RL and emergence, which can self-optimise tracking energy costs via collaborating intelligent agents. The case of robots with binaural microphone setups tracking down specific sound sources with minimal energy consumption in disaster scenarios is one such prominent example.

Chapter 3

A Distributed Framework for CASA

3.1 Introduction

The first chapter is centred around the design and development of the proper EDS that can support these tasks. Through the research gaps and opportunities identified through the literature review of this interdisciplinary study, it became clear that: (a) there are benefits to utilising microservice-based MAS with capability for designing and implementing socio-cognitive traits and social interactions, and (b) that is something missing from CASA distributed systems. Moreover, the system should be able to both be deployed on real devices and act as a simulation framework, too. The designs and implementations, with new ideas and utilising robust tools to achieve the goals, aspire to contribute a new tool for CASA research and development. But most importantly, to serve as the framework for the rest of the study that focuses on the tracking problem to be tackled here. The chapter focuses on presenting these designs and implementation details for the proposed EDS framework, demonstrating the way the viability of the components has been evaluated and discussing how they can help achieve the goals.

3.2 Conceptual System Architecture

The conceptual architecture of the developed EDS for CASA is on a foundation of microservices, enabling a highly modular and scalable framework for intelligent agent interactions. Each agent within the system operates as an independent computational entity, leveraging microservices to perform specialised tasks, and specifically signal processing (from a simulated environment) and decision-making on the intelligent nodes. By ad3.1opting a microservices approach, the system ensures that each functional component remains loosely

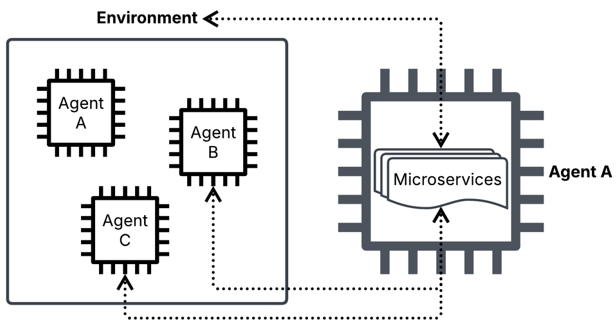

Figure 3.1: Conceptual interactions between agents, and between an agent and its environment, through the microservices at the core of each agent.

coupled, facilitating adaptability, and the ability to integrate new functionality as needed (e.g., decision-making strategy, signal-processing solution) for the specific CASA application.

This architectural choice also supports assorted distributed deployment configurations, allowing agents to be deployed across heterogeneous devices, including edge devices, cloud environments, and hybrid configurations, ensuring optimal performance based on the needs of the application. The primary contribution of this system, apart from serving as the means to conduct this study, is to also provide a framework and developmental approach to realising close to real-world applications in CASA. Such systems will be infused with all the benefits of a microservice-based approach that has not been explored in this research field, yet, at least when combined with intelligent agents in distributed systems as elicited from the literature.

The intelligent agents at the core of this system interact with each other, as well as with the environment itself, through these microservices (Figure 3.1). These microservices and their interactions will be described in detail when reviewing the designs of each component below in this chapter. However complex the implementations and interactions of these microservice-based components can be or become, there is a very high level of abstraction that can define families of such microservices and other system-level components that are used. How they interact among themselves is sequential and thus can comprise three different layers by which anything coming from the outside world passes through the intelligent node and results in the effect it has on the world in turn.

These three layers, which are depicted in Figure 3.2, are:

1. *Perception* – microservices that analyse anything received from the environment (or the entities within it).

2. *Intelligence* – microservices that employ results of above analysis to make decisions on how to act.

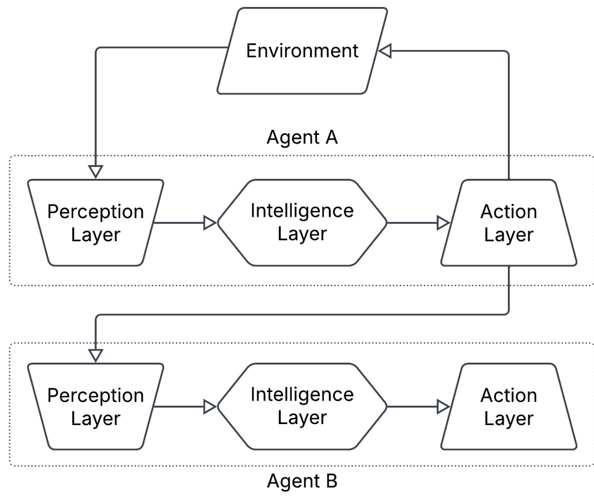

Figure 3.2: Conceptual interactions among the layers within two agents and their environment.

3. *Action* – microservices that implement the decisions taken, thus affecting the environment (e.g., move, communicate).

At the *perception* layer, agents employ CASA techniques to analyse and interpret auditory environments, extracting meaningful information from complex acoustic scenes. This involves CASA processes executed within the microservices framework, but also minor processes such as obstacle detection and listening for other agent communications at the later stages. For the EDS framework developed, the ability to distribute these services across multiple agents, and consequently devices, enhances the system's efficiency, enabling a form of collaborative processing of the acoustic scene, albeit without a common processing system for these results – the collaboration emerges from the agent interactions instead.

The *intelligence* layer is designed to accommodate assorted agent architectures that constitute strategies for solving the problem, allowing flexibility in the decision-making process. Agents can employ rule-based systems, machine learning models, reinforcement learning strategies, or hybrid approaches, depending on the application requirements. This adaptability is achieved through an abstraction layer that allows seamless switching between different intelligence paradigms without disrupting the overall system's operation (e.g., keep same perception and action layer but replace intelligent only).

At first glance, the *action* layer is the most straight-forward in design and implementation, given that it generally consists of simple actions dictated by the decisions made earlier, such as the robot moves, or it waits to listen better again, or it shares its information with other agents. Nonetheless, this layer is the one that has the most impact in the end for the problem at hand: every action costs energy and time and the sum of these costs are what ultimately leads to success or failure. This layer also handles communication, so agents can interact and collaborate. Naturally, via this layer agents can share learned insights with one another, forming an emergent intelligence network where distributed agents ultimately collaboratively refine their perception and decision-making capabilities over time.

Finally, the system supports diverse deployment configurations to meet varying computational and infrastructural demands. Agents can function autonomously on low-power devices due to the development technologies employed, collaborate in clustered environments for higher performance, or integrate with cloud-based resources for large-scale data analysis if needed (though not for the current study). This holistic architectural approach positions the system as a versatile and scalable solution for a wide range of CASA-driven applications, a contribution to interdisciplinary studies in the fields involved.

3.3 Core Components

This section presents design and implementation details regarding the core components of the developed EDS CASA framework. These components are essential for utilising the EDS as a framework for developing assorted CASA applications with intelligent agents powering the reasoning and functionality in the system. Naturally, these are not constrained only by the existing CASA problem that is being addressed, neither by the specific agents needed, and serve to establish the base of the distributed system (how nodes interact, how the find each other in a decentralised manner). Moreover, it gives a prototype implementation of simulation instead if needed around the *Environment* microservice that will be presented.

3.3.1 Interaction microservices

These interaction components – discovery and communication – of the system are the core of the distributed architecture. They allow assorted heterogeneous systems to achieve interoperability and (e.g. a mobile phone receiving streaming video and casting it to a TV) are an essential component for emergence to arise among entities that can communicate in a specific manner. That by no means disallows emergence to occur outside of communication, naturally, as the intelligent component can interact with the environment itself and cause it. In the case of a distributed system, there are two hard requirements that need to be satisfied: (a) a means for nodes to communicate with, and (b) a means for the nodes to discover each other so that they can communicate.

One of the benefits of the minimal API architecture is that the requests and responses can be performed in an asynchronous manner [164]. This means that the system does not need to be locked down while waiting for a reply and keep performing its functions. As such, an agent can send a request for communication and keep performing its tracking functions no matter if a reply is received in a timely manner or not – when it arrives the agent can act accordingly. With the current architectural approach, the minimal API implementation is designed with REST in mind. As such, the design pattern is as follows: an Endpoint maps an Action request to a function in a Service of the system (microservices in this case).

As an example, the following are synonymous:

- `POST > ‘‘user@provider.com’’ > http://192.128.236.10/users > IUserService.CreateUser(‘‘user@provider.com’’)`
- "`192.128.236.10`" create for me a new user whose email is "`user@provider.com`".

Communication The `ICommunicationService` interface composes the microservice handling API requests and responses in a RESTful manner for communication purposes. At the first iteration of the system architecture, its `GetStatus()` and `SendStatus()` functions are mapped to the corresponding endpoints and interact with another microservice of the system, `IStatusService`. Notably, the former is a stateless service whereas the latter is a stateful one, achieving a clear separation of responsibility and adhering to SOLID principles. The information delivered via the status service

The original supported Endpoints for this microservice are the following:

`GET /status` Returns the status of this node. Always available.

`POST /status` Receives the status of another node. Always available.

Discovery The `IDiscoveryService` interface defines the way the nodes in the distributed system can discover each other. Nodes can be configured to listen to a specific port at their defined endpoint in the overlay network they reside, and other nodes can try to hit IPs with requests that they know could be available in the system. An example would be the overlay network being configured to use the subnet `128.198.256.X` where `X` can be the number of the node. Here the agents can utilise the `DiscoverNodes()` method to send `GET` requests to the Endpoint and hopefully find new nodes in the system.

The default implementation for this microservice comes with preconfigured range of such IP addresses so that there is less time and processing cost involved into discovering nodes in the environment, as eventually will be dictated by the tracking strategies. As the study progresses, this can be extended to use a different implementation when the discovery response to another node may include the nodes that the system currently is aware of. In this manner, a social network can be formed from which interesting interactions may emerge during CPS attempts.

The default endpoints for the discovery microservice are:

`GET /` Returns the contact information of this node.

`POST /` Receives the contact information of another node.

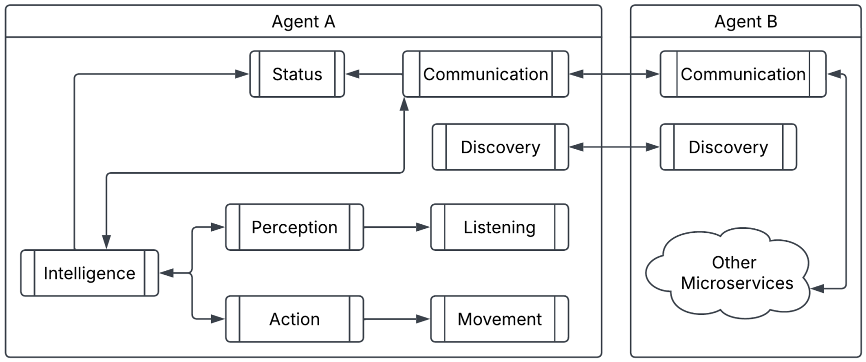

Figure 3.3: An example of interactions among the core microservices of an agent, and with those of other agents. Interaction directions describe which microservice can call methods from another.

3.3.2 Intelligent agents microservices

The bio-inspired trackers, the intelligent agents of the system, comprise the most complex of its components, utilising the hosted worker architecture and injecting all services provided up until now, including the ones that perform the communication functions, thus capable of networking. This creates a list of dependencies including communication, status, discovery, movement, and environment. To be able and realise their goal of locating a mobile source, however, they also depend on perception of the environment and taking actions – as is the case for AI nodes operating under the RL machine learning paradigm (interaction with the dynamic environment).

The default implementation for the new microservices will be discussed in greater length in the subsequent sections, as they heavily touch the realms of CASA and RL respectively for perception through listening and decision-making. A teaser for the new services introduced at the abstract design level is found below. The idea of having this type of abstraction is that so it will be easy for architects modifying the system to their needs for future work and assorted real-world problem-solving studies and applications, to be able to simply add their own interface. If a future study attempts to add visual aid for the robots, all that is needed is to inject it in the perception layer and call it when the update function is called. The same for acting if the robot can fly, or pick up and put down things, or even make radio calls to support personnel.

The possibilities are endless and all that will be needed is to simply add the interface with the implementation. This provides a highly modular design that can create unique applications. The interactions among the primary agent microservices that exist in the system are illustrated in Figure 3.3. It is evident that the *perception* microservice can, for example, be expanded by a *vision* microservice that can visually capture information from the environment, and consequently the *intelligence* microservice can then factor that in reasoning. This is one of the strengths of this modular architecture.

The final list of the dependency tree of microservices for the trackers is formed as:

- `IDiscoveryService`
- `ICommunicationService`
  - `IStatusService`
- `IIntelligenceService`
  - `IPerceptionService`
    - `IListeningService`
      - `IEnvironmentService` (simulations only)
  - `IActionService`
    - `IMovementService`
      - `IEnvironmentService` (simulations only)

3.4 Simulation Components

The purpose of this section is to explain the details of how specific components crucial to this study have been developed. These mostly pertain to the simulations that followed with the strategies that were developed towards solving the energy-efficient tracking problem at hand. The focus is on the process of creating artificial treefrog mating calls close to normal ones that can be used as sources for creating the localised sounds the mobile sound sources use, to capitalise on cues known from the treefrog studies that can serve as tracker preferences. Then how the acoustic scene has been created for accuracy, including the immense dataset of IRs for generating the localised treefrog sounds. Finally, the SSL method that the trackers employ is detailed, with elaboration on how it was tested for accuracy to ensure that the strategy experiments to follow were carried out correctly.

3.4.1 Environment microservice

The representation of the physical environment is an essential component of the system when it will be used towards research through simulations. There are arguments to be made towards incorporating it in the solution as node that can communicate this with accuracy, but this adds another vast dimension to the real-world application of the system and its intended purpose. Specifically, the nodes would need to possess their own system for determining their position to report it, or for the environment node to be able to determine their positions. There are trade-offs here with significant costs to computational power and energy consumption, and the focus of this thesis on attempting to solve this problem through a different, bio-inspired manner, does not align with this approach.

In the form of a stateful microservice, the `IEnvironmentService` holds key information pertaining to:

- The location of the sound sources.
- The location of the trackers.
- The boundaries of the environment.
- The number of obstacles.

Regardless of the complexity and dynamic nature of the real-world problem, the simulated environment need only be concerned with the aspects due to the manner by which a distributed system operates. The interface of this microservice offers the means to both trackers and targets to update their location when they move. The environment then responds to this update by letting the nodes know, for simulation purposes of course, whether they have moved into an empty space, a target, or an obstacle. The default implementation keeps this state in-memory and not in an external system but can be parametrised to do so with efficient persistence systems – or costly such as databases – as needed for the application.

The simulated environment For the purposes of this thesis, the simulated environment will be a two-dimensional grid with a size of 10 tiles for each side, each tile representing $1m$, essentially a square room. More details for this choice will be presented when discussing the binaural simulation for the room and the generation of IRs. Finally, the entities moving inside this logical 2D space will be able to move on all 8 basic directions of the compass (N, NE, E, SE, S, SW, W, NW). Naturally, movement implies that no obstacle is present, otherwise the service will return the corresponding result. Entering the space of an audio source signifies that the tracker has encountered an obstacle, too, which they need to handle in a proper manner in their decision-making process.

The generation of obstacles needs to be addressed. These obstacles represent points where the moving tracker cannot cross and bumps into, simulating possible debris, walls, or even furniture in a room or floor. A way to ensure that the trackers can indeed reach their target is of the utmost importance. Attempts at fully randomising the allocation of obstacles was found to create walled off sections with targets or trackers trapped inside. Maze algorithms were studied next due to their unique property of always having a path through all corridors and from start to end, albeit most algorithms produce an extremely snake-like result that cannot represent the expected scenes faithfully.

Recursive division is the only algorithm that could develop something that even remotely resembles the intended acoustic scene, due to its tendency to create mazes with long straight walls and corridors. Indeed, Figure shows how an empty room can be turned into a maze in just a handful of steps but stop at

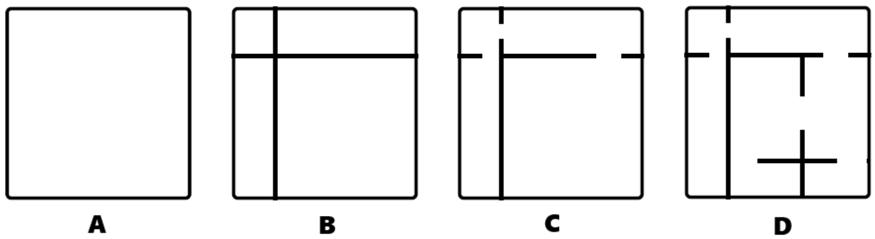

Figure 3.4: The recursive division algorithm in action: (A) initial empty environment, (B) divide by two walls at random (x, y), (C) create holes in the walls, and (D) recursively repeat B-C until necessary.

one of the previous steps and there is something closely resembling an apartment or building floor. The implemented algorithm can produce two types of rooms, by stopping at a different step each time and randomly removing a random small number of obstacles already added: one that has few obstacles (e.g. step 3 with 3 tiles removed), and one that has many (e.g. step 4 with one more division and 3 tiles removed). Iterations required for experiments can be found in Appendix A.

The mobile sound sources Comprising the targets of the trackers in the system, similarly to the environment they are not designed as nodes, too. There is no permutation of the energy-efficient mobile tracking problem where these components would need to be real targets outside of simulations. Consequently, their operation is not taxing to the system with an extremely lightweight implementation as they do not need to even keep their own state in memory. They do depend on the environment microservice, however, so that they can update their location when they decide to move, but only through the `IMovementService` that injects it. This service is introduced here and can be used by any component or node of the system that needs to move (i.e the trackers). It has the simplest interface of all the services, with the function `Move()`, which in its default implementation for the simulations merely updates the location of the entity and its head orientation in the simulated environment. A variant of this microservice for a real-world application would be akin to calling the API of the robot to move its wheels (e.g. a low-level function in an embedded system to spin the rotor).

At this point there is an encounter with another elegant solution the lightweight .NET services offer for representing repeated tasks (i.e. the target moving at set intervals), but which also generally characterise intelligent agents and AI components. Thanks to the Hosted Services the mobile sound source component can be declared as a worker that awakens at discrete points in time with a set interval and perform its tasks (e.g. cleaning a database table) – or that can even run continuously if required. These types of hosted workers can perform

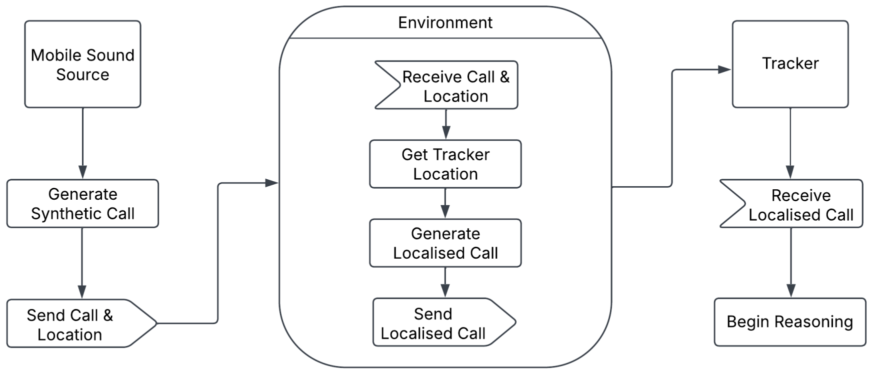

Figure 3.5: Demonstration of mobile sound source output propagation to the tracker via the *Environment* microservice. The mobile sound source is inspired by the male treefrog, whereas the tracker by the female treefrog from biology studies.

from menial tasks (such as for the mobile sound source) to extremely complicated, such as interaction with other APIs or running a whole distributed system deployment for big enterprises.

For the task at hand, the worker wakes up at set intervals and decides if they will move randomly or not. The major advantage over treating them as simple classes is the asynchronous mode in which they can be set up to operate, effectively freeing up resources in the system when there is no need to perform an action. Given that workers consume resources only when waking up, they manage to free up the node to perform other tasks if needed. This provides both potential for more intricate applications with the resulting distributed system, where the nodes perform multiple functions, but also manages to conserve energy for the tracking nodes that will be discussed next. Finally, with regards to the audio generated by the mobile sound sources that the trackers will attempt to localise, the details are presented in the section of the chapter for the auditory sensory component.

This audio, which the tracker should hear localised, is propagated with the mechanisms featured in Figure 3.5. How these sounds are generated and localised within the simulation environment is detailed in the following sections. The *Environment* microservice is there for simulation purposes, so if a real-world application of the system was deployed, this microservice should be removed from the interaction pipelines. Indeed, the mobile sound source depicted there would be an actual entity in the real world that generates sound in whatever way it can (naturally or artificially), whereas the tracker entity would have actual microphones that interface with its reasoning layer providing the already localised call carrying all the properties of a real-world environment applicable.

3.4.2 Generating treefrog sounds

With a view to having numerous audio sources moving in the environment, each one with different characteristics so that the agents may have decisions to make towards tracking the right one, different treefrog audio sources would need to be available. However, for the purpose of repeated experiments it is not feasible to acquire many actual, clear treefrog calls that have the exact desired acoustic properties that will be set for the trackers, too. Consequently, a call generation mechanism had to be implemented that can incorporate these parameters. The signal processing toolkit of MATLAB has been used to produce synthetic sounds emulating those of the treefrogs, based on a set of properties that define the synthetic call, and by empirical testing the output by ear compared to a real-world treefrog call during generation.

The properties that are passed as parameters to this function are:

- Attack time (ms).
- Decay time (ms).
- Gap time in (ms).
- Start pitch (Hz).
- End pitch (Hz).
- Harmonics number.
- Harmonics levels.
- Number of pulses.

Values for these properties passed to the function and as used in experiments can be found in Appendix A.

The synthetic call generation process is the following:

1. The attack and decay ramp times are calculated.
2. An envelope is created using attack, decay, and gap time.
3. Envelope is made complex using harmonics parameters.
4. The signal is then scaled to peak value using harmonics levels.
5. A final waveform is produced by using the number of pulses.

To evaluate the efficiency of the outputs of the synthetic call generation process, an ideal recorded sample from a marsh environment has been used as a guide towards fine-tuning the property ranges. The resulting signals have been visualised and compared to the actual calls, with the results in Figures 3.6 to 3.8, empirically verifying an approximation of the expected outcome.

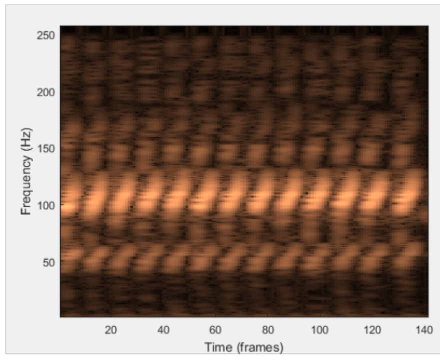

Figure 3.6: Original treefrog mating call obtained online from currently unavailable source.

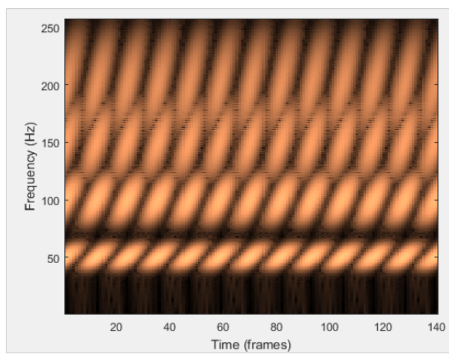

Figure 3.7: Synthetic call produced by the auditory sub-system for a higher-pitched treefrog call.

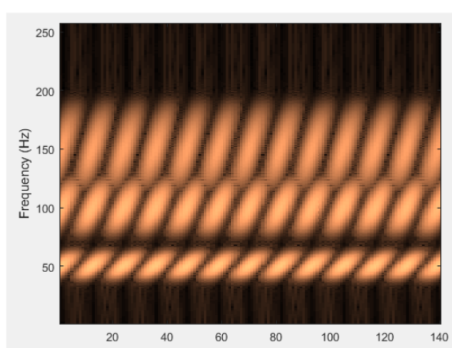

Figure 3.8: Synthetic call produced by the auditory sub-system for a lower-pitched treefrog call.

3.4.3 Generating localised sounds

This crucial step in the design and implementation of the CASA subsystem involved several intricate steps to bring to fruition. In a real-world application of the system, this part of the work would be entirely discarded due to the input being provided by actual sound sources. This was an iterative work that had to tackle several issues that arose during implementation and the steps to produce the localised sounds had to be repeated several times. Problems faced were unexpected results in localised sounds due to problematic settings (e.g. erroneous high-volume reflections when close to walls) or problematic algorithms under certain conditions (e.g. reverberation applied when using values above a specific threshold) in the binaural simulator. Also detrimental was the immense localised collection size and processing power to create the multiple IRs, as well as the availability of the dataset for running experiments (i.e. the generated impulse responses for the simulations).

Two steps were necessary to complete this task:

1. Design the acoustic scene and its characteristics, imposing necessary constraints.

2. Create the necessary HRIRs that would provide the localised sounds at runtime.

3.4.4 Acoustic scene simulation

The acoustic scene is the environment wherein the bio-inspired trackers will live and attempt to fulfil their purpose. Real-world parallels to the agent tracking a sound source moving in the environment cannot be captured in their entirety without large-scale efforts. This fact dictates that several constraints need to be imposed to the acoustic scene, while at the same time ensuring that these limitations will not become an impediment to the goals of the experiments. But to be able to describe the acoustic scene, it is necessary to image an analogous real-world problem scenario where energy-efficient tracking of the audio sources can provide a solution. To this end, and for the future experiments, the parallel drawn is to that of an area in a disaster scenario with zero visibility and people calling for help trying to find their way (e.g. a subway tunnel collapsing and no electricity, a building on fire and thick smoke) and debris littering the floor, where robots are deployed and need to locate their targets through sound. This will be revisited when the iterative experiments are presented, too, but an early reference here is required to explain the specifics of the acoustic scene.

To start with, the "shoebox" binaural simulator supports three-dimensional (3D) rooms expressed as 2D grids stacked on the z-axis. A 3D environment where robot-like devices operate tracking targets implies a much more complex environment and problem to solve – from devices akin to flying drones, introducing the extra problem of extreme self-noise, to totally different energy costs, and even the problem of how such devices could be controlled without vision or a similar system (and if there where such capabilities, why rely on sound alone).

These concerns and issues outside the scope of this research undertaking led to the decision of the modelling of a 2D space pertaining to the movement and obstacle design, where traditional robots on the ground are simulated trying to localise and locate their target. The major advantage from the 3D capabilities of the simulator is that the sound can be simulated in 3D while the operating space can stay in 2D, therefore giving a more accurate representation of the sound signal. The base input model used was that of KEMAR readings for anechoic room [224].

The dimensions of the room in metres have been set as a 10 by 10 on the 2D space, with a height of 3 that could represent a large building floor for the scenario. The permutations of listener-speaker positions and orientation can grow to high numbers as will be showcased next, constituting another reason for this. Reverberation has been set according to the standardised formula $RT60 = k \cdot \frac{V}{A}$, where factor k is $0.161m$, V the room volume in m^3, and A the equivalent absorption surface in m^2. A calculator has been utilised to provide the proper value, which operates on material input for the surfaces (i.e. painted concrete) and dimensions of the room, also providing the related absorption and reflection coefficients for the simulator [225]. Head orientation of the actors as per the simulated environment configuration adheres to the 8 base compass directions, which have been mapped to $0 - 315$ azimuth, where 0 is North and 315 Northwest, on the proper intervals of 45. As a clarification, this is listeners with human head physiology (from KEMAR) tracking treefrog sounds (from synthetic calls), which is beneficial due to the ease of modelling the bio-inspired strategies based on the known cue requirements for treefrogs, and for developing and testing the localisation solution. These parameters can also be found in Appendix A in more details, bound to their framework values.

3.4.5 Impulse response generation

Having set up the acoustic scene, the next step is to generate a proper IR. The system requires the following inputs for both the listener-tracker and the speaker-target: position in x-axis, position in y-axis, and head orientation. Once these have been provided, the algorithm can be run to produce the proper function for the HRIR on the side of the listener. The output can then be convoluted with the synthesised treefrog call and subsequently analysed by the localisation function at the tracker. While this approach provides verified, excellent regards, the time and processing requirements for generating the single IR for a specific pair of listener-speaker locations with specific head orientation, requires a few minutes. Evidently, it would be infeasible to do this at simulation runtime for every tracker. A solution to this obstacle is to pre-generate the impulse responses. Once the responses are stored, the convolution of one with a synthetic call is a matter of a couple of seconds at most. This would significantly reduce the time needed to perform the function on the fly as needed by the tracker and make simulations feasible.Nonetheless, even creating all the needed files in advance is a lengthy task.

Attempting to calculate the required responses provides the following insights:

- Potential listener or speaker single locations: $10 \cdot 10 = 100$
- Potential listener and speaker locations pairs: $100 \cdot 100 = 10,000$
- Potential listener or speaker head orientation on a single location: 8
- Potential listener and speaker head orientations on a single location pair: 64
- Potential listener and speaker head orientations on all location pairs: $64 \cdot 10,000 = 640,000$

A total of $640,000$ HRIRs need to be generated to cover this scenario, a function of $IR = (x \cdot y \cdot o)^2$, where x the width of the room in metres, y the length of the room in metres, and o the number of available head orientations. This is one of the reasons that imposed the constraints on final environment dimensions or orientations – a mere increment of $1m$ could lead to tens of thousands of additional files. Moreover, the generated audio file to be used for the convolution with the synthetic call has a size of $200KB$, which results in a dataset of the size of approximately $130GB$. Notwithstanding, the most important problem of generating these files was time. At the moment in time when this endeavour took place, it took an average of about half a minute, or $30s$, for the local machine to create one. This could lead to cost in hours in the region of 5000, or roughly 7 months of non-stop IR generation.

To make this feasible, generation of datasets was moved to the first generation HPC clusters at the University of Sheffield, which supported MATLAB functions and could handle the workload of the Two!Ears binaural simulator. The machine used to create the original batches of datasets was the – now retired – Iceberg cluster [226]. Requiring several milliseconds for each file compared to half a minute, the file generation costs were reduced by a factor close to 100, making this task feasible. The restricted personal storage space on the cluster for queued job outputs required breaking down the job to smaller ones, specifically the manageable case of rows by orientations. Iteratively the files where created, downloaded over File Transfer Protocol (FTP), storage cleared, and new rows with orientation queued, over numerous iterations.

3.4.6 Binaural localisation at the tracker

The final part of the CASA component in the system is the localisation of the input signals at the side of the tracker, which refers to the default implementation of the `IListeningService`.

The auditory pipeline for the experiments can be defined in the following stepwise manner:

1. Generate a random synthetic call akin to treefrog calls for each mobile speaking target (mimics a speaking target).

2. Perform a convolution of the synthetic call of each target with the proper HRIR for the locations and head orientations of both tracker and target (mimics sound reaching the ears of a listener)

3. If more than one speaking targets, then merge the outputs for each one into a single signal (mimics a complex acoustic scene).

4. Pass the signal through the binaural simulator so that the tracker can derive an angle in azimuth of the desired target (performs localisation on the provided acoustic scene).

Step **1** has already been detailed, while the rest are discussed here. In a real-world setting, there would be no need for the extra step of combing all sounds as in the simulated environment to create a faithful representation of it. Each channel of the stereo signal output of the previous stages (i.e. a channel for each ear) is convoluted with the synthetic call to create the expected sound. A scaling factor for normalisation is applied to minimise the artefacts introduced by the convolution **2**. Lastly, mixing of any noise and call signals that essential to simulate the real environment performs on the input are performed at the beginning of this stage, to simulate what the tracking agent would be hearing in reality, albeit with the limitation of excluding self-noise. This is step **3** and it is achieved by first adjusting the two different source sound levels ensuring balance in volume (normalise to prevent clipping) and subsequently combining them with a simple addition (superposition) without any weighted mixing.

The final step **4** is the crucial one for ensuring that the core entities in the experiments, the bio-inspired trackers, properly perform the localisation tracking so that the focus can shift on the actual tracking behaviours. The tracker must be able to perform two tasks: (a) determine the best target in the environment to follow, and (b) to localise that target. Task **a** is a highly involved process for the treefrogs, with biological studies discovering that they tend to show a preference to the overall envelope of a treefrog call, rather than very specific acoustic cues. However, there are two secondary cues that are play an important role to treefrog decision-making and can be exploited instead of attempting to do complex envelope matching. These are the average frequency, especially when it falls within the range $1000 - 1300Hz$, and a higher pulse rate. The weights for each are not clear, so the algorithm is designed to pick the one with the least deviation from the preference set on the tracker. Accordingly, MATLAB signal toolkit functions are utilised to find the local peaks in the signal to determine the pulse rate – peaks over the average treefrog signal period as defined in the synthetic call generation – and to estimate the frequency. But for the tracker to have a choice of which source to choose, they need to have a list of potential sources first.

This is where the localisation pipeline acts:

1. Perform auto-correlation (ACF) for each ear to estimate *frequency* and *amplitude*.

2. Load the HRTFs for each ear on the location.

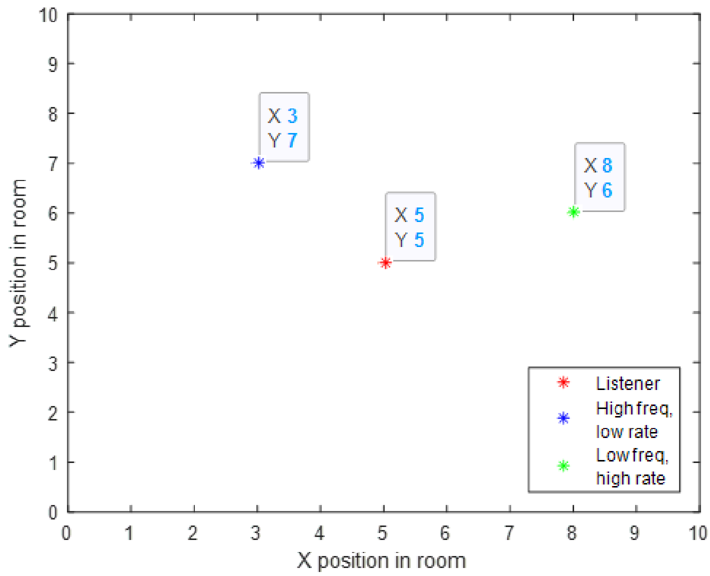

Figure 3.9: Overview of a sample room performing tests with the CASA pipeline. Listener is -45^o from the high-frequency target ($1100Hz$), and $+72^o$ from the low-frequency target ($800Hz$).

3. Iterate through *azimuth* and each orientation (North to East, North to West) computing the *ITDs* with cross-correlation (CCF).

4. Generate a *mapping* function for *ITD to azimuth*.

5. Use CCF on the input signal and apply weighting function by ACF to find the ITDs and *get the relevant peak(s)*.

6. Employ the HRTF azimuth mapping function to *get the azimuth for each peak*.

7. Choose among peak azimuths based on frequency and amplitude *deviation from preferences*.

A sample demonstration of parts of the localisation pipeline and tests against the ground truth can be observed in Figures 3.9 to 3.12. To evaluate the accuracy of localisation, two tests where run: (a) complex scenarios to ensure that the proper target was chosen each time, and (b) a full test of all returned azimuths compared for a listener in a static spot and a speaker in every other spot of the grid. For the evaluation of test case **a**, a handful of scenarios were created where the listener had two speakers, one with lower frequency and pulse rate and another with higher values for each. The full pipeline for generating the acoustic scene was run and the chosen azimuth was tested against the ground

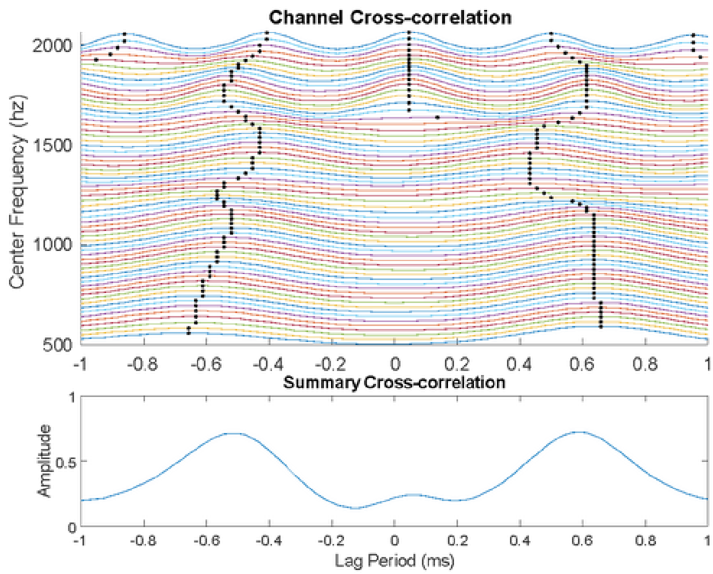

Figure 3.10: Cross-correlation and summary, targets estimated around -55^o (-10^o error) and $+78^o$ ($+6^o$ error).

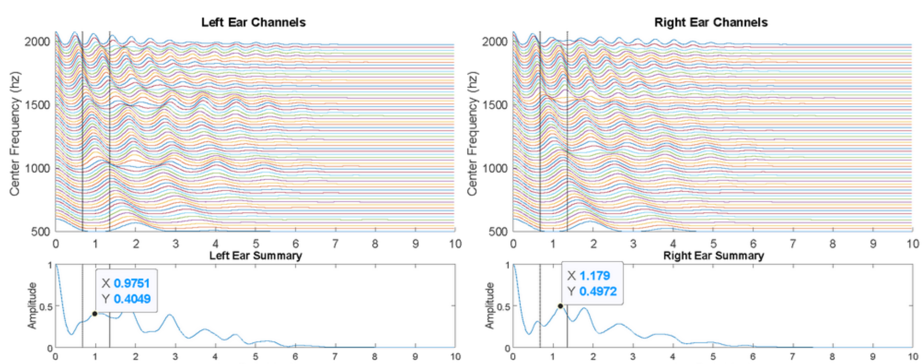

Figure 3.11: Auto-correlation for both ears. Left ear estimates a $1025Hz$ target ($-75Hz$ error) and right ear estimates a $868Hz$ ($+68Hz$ error).

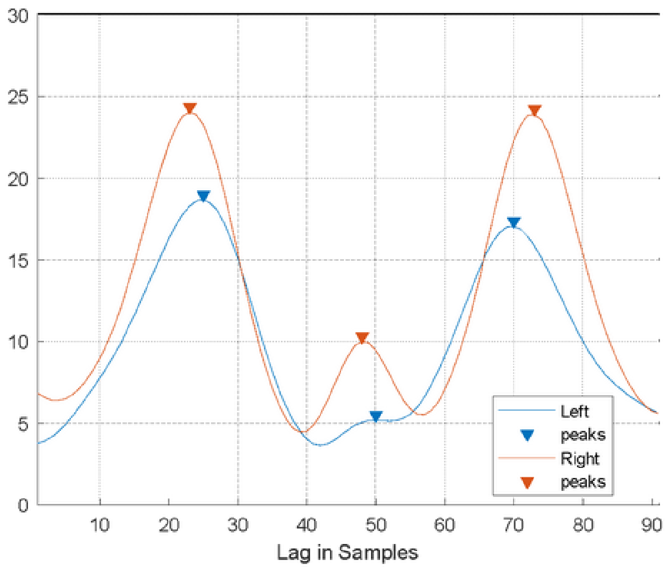

Figure 3.12: Weighted auto-correlation using period snapshot from ears applied to all cross-correlation channels, keeping the two highest peaks for each ear and using their average for result.

truth. This resulted in a near perfect match for the expected choice, the outliers falling to the category of $\pm 100Hz$ max.

Regarding test case **b** the results were extremely promising, and they were tested against the ground truth for both lower and higher frequency sources. In fact, the average error was $\pm 6^o$for frequencies around $1000Hz$ when the distance was up to $8m$, compared to $\pm 10^o$ for for $1300Hz$ for the same distance. These values fall indeed within the $\pm 10^o$localisation accuracy that treefrog studies have shown, and which are also like human capabilities [35]. Regrettably, beyond these distances the margin of error approximated the class of 40^o. This can be accredited to two factors, firstly that the signal has degraded due to distance, and secondly that the listener is very close to a wall and confused due to reflections (range of $8 - 10m$). This could constitute a significant limitation to the study, were it not for the fact that this allows for exploring how efficient the tracking strategy overall can be when it cannot get the best results out of the CASA component under adhere conditions.

There is one important caveat when using the measured ITD approach, however, and that is the front-back localisation problem. This problem refers to the fact that a sound arriving from either the front or the back of the listener, same distance and angle, will always have an identical ITD, as evidenced by *Step 3* of the localisation pipeline above. There are typically two ways by which binaural biological organisms attempt to solve this problem: (a) use a complementary sense – predominantly vision – and, (b) turn the head a few degrees to one side so that the distance changes [227]. ILD has been considered for solving this problem in recent studies, however the results indicate that there need to be very specific conditions in the acoustic scene for the ILD predictive model to work [228], which is uncharacteristic of the problem this thesis aims to solve.

This, naturally, settles the design choice down to finding a way of getting a better ITD reading, such as by rotating the head by at least one whole compass orientation (e.g. $\pm 45^o$ to spend less energy on rotation) towards the ITD-suggested azimuth so that a better reading can be gained. The unfortunate by-product of this design is higher energy costs, specifically the robot having to use motors to turn the head with microphones, as well as time costs. This is not an issue that should affect the result of studies when working with the original bio-inspired treefrog strategies, because treefrogs have nearly no reliance on vision unless they have another treefrog in front of them, and especially in marsh or rainforest environments with high vegetation where it is nearly impossible to see. An alternative to the head rotation is standing still and waiting for the next call to come, hopefully by the target having moved. However, some targets may be stationary for extend periods of time as they wait for someone to reach them, thus more time could be wasted in locating them – a reason why treefrogs mainly attempt to solve the problem with head rotations.

Having finalised this component, **Objective A1-O2** has been brought to completion.

3.5 The Bio-inspired Trackers

The most important entity in this thesis is the tracker – their capabilities, behaviours, and interactions are what will shape the outcomes. The entire distributed system being developed, the design of its microservices, as well as the elaborate CASA subsystem, have all been to enable the bio-inspired trackers to fulfil their purpose. While their designs are more detailed and documented in the respective sections in the following chapters, starting from the *Reactive* bio-inspired strategies, then moving to the *Deliberative* architecture of the *combined* strategy, thus ending with the *RL-based* strategy of the *adaptive* and *socio-cognitive* strategies, an introduction to this component is here as the baseline for developing intelligent agents in the proposed EDS.

All the components so far have been showcased, except for the last one: the decision-making process, which is the implementation of the microservice `IIntelligenceService`. On a higher abstraction layer, this service is responsible for calling when needed the `IPerceptionService` to develop an understanding of the local state of the world for the agent and then process this state to determine which action to take through the `IActionService`. The latter will only have a single possible option at this point in the design and that is movement, much like the `IPerceptionService` can only rely on listening. Nonetheless, it is provided for future work that could have the agents, and by extent possible robots in real-world applications, to perform more complex functions if

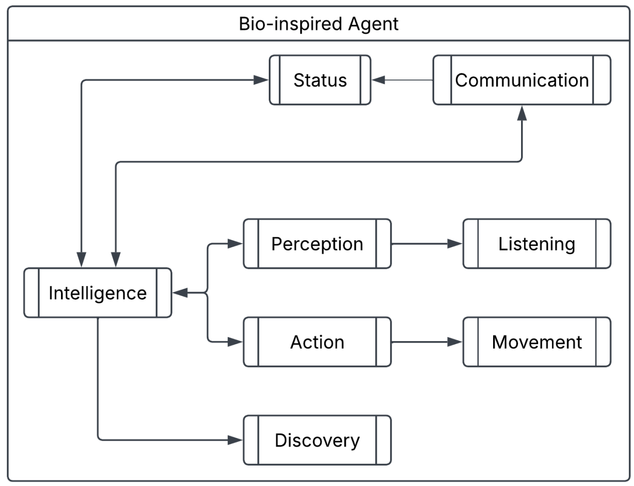

Figure 3.13: The preliminary microservice interaction designs for the bio-inspired intelligent agents, serving as a CASA-powered entity in the system (arrows indicate invocation capability).

needed (e.g. move and pick up something and then notify someone).

From the new microservices added, perception is labelled as `IPerceptionService`, which is dependent on `IListeningService` that provides clues pertaining to that sensory input. This listening component attempts to emulate a captured sound that will be localised, which for the goals of this project will be supplied via the binaural simulator. What perception attempts to do is understand the local state of the world (i.e. the state limited only to what the agent can understand) by relying on inputs from other functions (i.e. listening for the purposes of this thesis), and returns this state of the world to the intelligence service (`IIntelligenceService`) in a format that can be processed decide how to act upon. The complex interactions of these microservices in the agent are depicted in Figure 3.13.

The decision-making process of the agent at this point in the design of the study is that of a simple *Reactive* agent architecture, which decides every few seconds, after evaluating its state and having received a stimulus. At the starting point of this research undertaking no more complex AI has been deemed necessary, because these capabilities are enough to model the two basic treefrog behaviours (i.e. regular, explosive breeders). As this project evolves, the complex balancing act between exploring the environment and exploiting gathered knowledge to achieve the goal, the core pillar of RL, will be introduced to this agent. The more abstract life-cycle of the agent can be described in the flowchart found in Figure 3.14.

With the design and implementation of this final component **Objective**

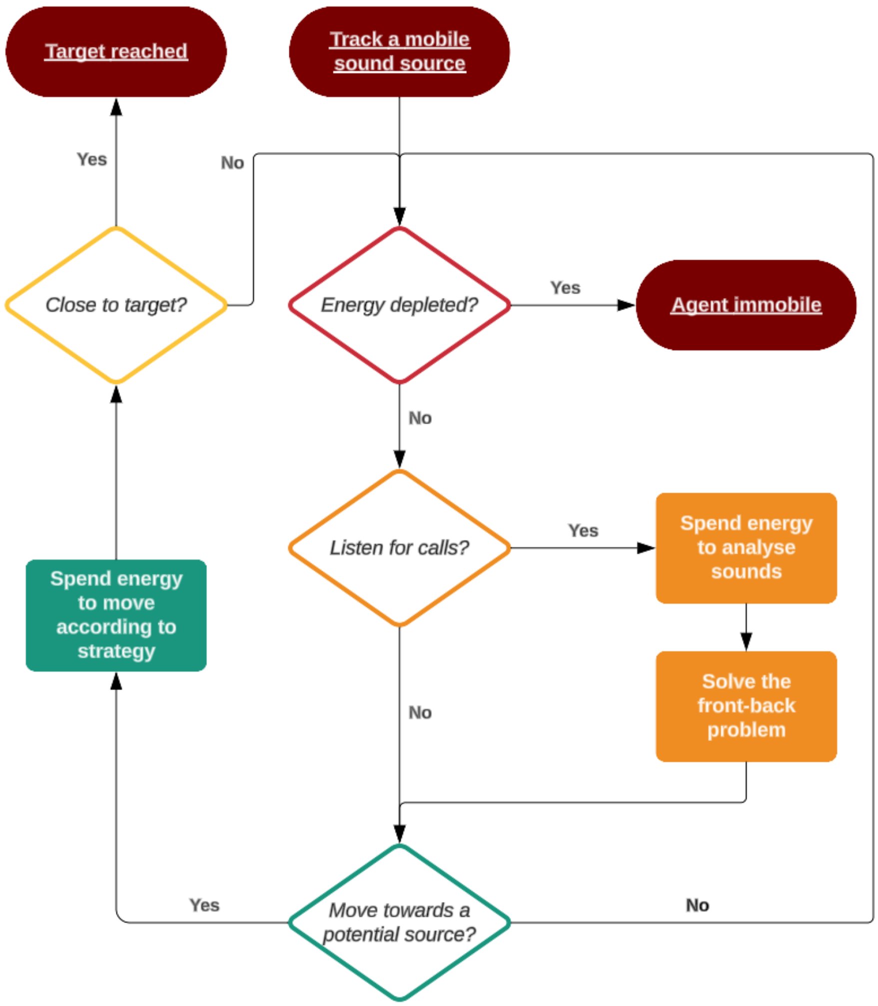

Figure 3.14: Basic life-cycle of the agent in a higher level of abstraction.

A1-O1 has been achieved.

3.6 Conclusions

This chapter has portrayed the design and implementation of a distributed system that allows nodes to communicate and discover each other effortlessly across heterogeneous devices, either strong or weak performance-wise. Furthermore, it has the architecture to perform a multitude of CASA functions on the nodes, such as generation of accurate synthetic calls and robust localisation of sound sources in the simulated environment. A core contribution of the new system is that it can hose host AI components that can make complex decisions and take actions as needed. The ease of extending these components for other types of intelligence and functionality will also be demonstrated in the following two chapters.

While a multitude of systems for CASA exist for each specific aspect involved, as observed through the literature review, it also became clear that not all of them attempt to combine all that this system attempts to achieve here and this is where the innovation lies. Microservice-based CASA system architectures have not been spotted outside the realm of sensor networks or similar solutions. Most importantly, they do not allow for the flexibility of possible deployment scenarios demonstrated through the architecture of this new system. Indeed, its development framework allows it to compete even with lower-level languages and frameworks, establishing it as a powerful tool for developing both simulation and actual systems that could benefit from these advantages. Meanwhile, most systems are designed to either run in simulations or are on actual devices, not easily allowing through their architecture for both with a mere different registration of a service.

All the above meld together to provide a system that might not excel at very specific, single-responsibility and application tasks, but it can tie them all together and hopefully serve as a framework for future research on the field of CASA. Accompanied by this document on how to extend it effortlessly, both as seen so far and with the following chapters, and with the documented process on how to generate datasets with robust tools that can be used to create scenarios. Several other frameworks exist, but the developed EDS here will show its strengths in the future within this study, how it can go so far as to model learning AI, or to operate with socio-cognitive traits and demonstrating emergence.

This might not be as significant a contribution to the field as the work that follows it, the evolution from the bio-inspired to the optimal strategies with emergence that solve the chosen tracking problem and given the numerous other CASA systems with higher expertise in application and performance. However, what it aspires to do, that is to serve as a tool for researchers that aim to perform similar research, with an ease of swapping between simulation and real deployments, or from one to another AI approach, is also the gap identified in the literature that this system attempted to fill – no system could be found through the existing literature that could tick all the boxes mentioned above and serve as the ideal tool for carrying out this study as envisioned. In closing, this now allows for a strict focus on contriving an energy-efficient strategy for tracking mobile sound sources next, through a framework for the future exploration of emergence, thereby achieving **Goal A1** of this thesis.

Chapter 4

Developing an Adaptive Strategy

4.1 Introduction

The next research endeavour towards bringing this project to fruition is the development of a tracking strategy for the bio-inspired agents, in particular one that can adapt to the dynamic environment of mobile audio sources and obstacles – and by extend a possible rescue undertaking in a disaster scenario in real-world applications. Adaptivity for AI and autonomous systems is an indispensable trait when they are put to the task of solving problems with a high degree of complexity. But for an efficient problem-solving tool to be assembled, concrete foundations to build upon are paramount. These are provided via the strategies that the treefrogs exhibit in real life and have allowed them to survive as a species for hundreds of years in extremely bio-diverse and threatening environments such as the Amazon Rainforest.

The concept of energy in the context of the problem at hand is also elaborated on, how it is addressed in nature and how it can be adapted to the system in both a conceptual and physical manner. The sections trailing after this analysis constitute the experiments ran using the system and the assorted tracking strategies, before concluding with the progress so far and determining the progress so far.

The research methodology followed is the same for each set of experiments, including consecutive chapters:

1. The concept and goals of the experiments is introduced.
2. The research questions are formulated.
3. The required modifications to the system are detailed.
4. The experiment designs, variables and methodology, are presented.

5. The results are gathered and analysed.
6. A discussion wraps up the set of experiments with thoughts on moving forward.

4.2 Energy in the System

This part discusses briefly the concept of *energy* in the context of this study and, by extent, in the problem of efficiently tracking multiple mobile audio sources in dynamic environments. This is essential to the design of the experiments with all strategies and their evaluations for the rest of this work.

4.2.1 The concept of energy

Energy is a concept that has been mentioned repeatedly throughout this thesis so far, still there has not been a formal reference on how exactly it is represented in the system and what exactly it entails in the context of this thesis. Consequently, when this thesis refers to the concept of *Energy*, it refers to the quantitative property, finite in quantity, of a node in the distributed system that can performing actions in the environment by consuming an amount of this property. For the goals of this thesis, energy thus refers to motor motion, either for head orientation changes (e.g., solving the front-back problem, obstacle detected) or for movement, but also for listening processes.

From the cosmological zero- or constant-energy-sum state of the universe theories to the prevalent theory of a heat death of the universe [191], energy is a fascinating concept and its interactions can be observed and felt all around and inside of humans constantly. In robotics and distributed systems, energy conveys the meaning from the prior definition in this thesis. In Biology, energy essentially dictates the very life of even the smaller living organism – the very core definition of living in biological studies is the continuous reproduction of cells through the consumption of environmental energy [229]. Management of incoming and outgoing energy is, naturally, the most integral aspect of life and how it is addressed provides either efficiency or impediments to every action that needs to be taken, every problem that needs to be solved.

As a result of the impact of energy management, biological organisms and systems are in a state of constant struggle to follow the principle of conservation of energy. This principle – or law – states that energy cannot be created or destroyed but merely converted from one form to another. When a system is treated as a closed system (i.e. no transfer of energy in or out), this principle implies that the total energy of the system remains constant. The volatility of a living organism due to its constant interactions with external systems, and by the definition of life itself, puts them in the peculiar state of always transforming energy from one form to another. Notwithstanding, the internal system of the living organism attempts to conserve the energy – it produces cells because energy needs to be expended to keep the system stable, or it consumes energy to replenish energy lost on performing an action [229].

Outliers always exist in every complex system, still the studies show that even the outliers have the firm belief that by expending more energy they will eventually reach a more desirable state of energy stability in the near or far future, whether they are hallucinating or not [230, 229]. The case of treefrog explosive breeding, is a prime example. They descended to the ground from the trees to make, abandoning briefly their arboreal state, and optimise every single action and choice they take towards spending the least amount of energy to satisfy their goal before returning to a more stable energy (i.e. arboreal) state again. In fact, some of the more explosive breeding treefrog species tend to do this only once in their lifetime, resorting to spend the rest of their life in a constant stable energy state and skip subsequent mating seasons [231].

In robotic systems the difficulty in energy conservation stems from the fact that they are typically designed to spend energy to perform a complex action [232]. Meanwhile, for most of applications they do not generally have the capacity to replenish their energy (i.e. battery) reserves, nor do they have the expertise to sustain themselves dissimilar to biological organisms thanks to years of evolution, or at least not yet. Consequently, optimising energy consumption to achieve the goals constitutes an imperative design goal in most such systems. As will be highlighted next and has been pinpointed in the domain exploration study, tracking applications tend to be mostly static listeners that primarily focus on minimising energy costs for the algorithms employed, or moving trackers that do not necessarily have the highest priority of reaching their target but rather to better understand the acoustic scene. This is a research gap that will be exploited in this thesis, with a view to developing a strategy that can address this problem and a framework for studying such systems.

4.2.2 Energy efficiency in the system

To produce meaningful results for the experiments in this thesis, there needs to be a mapping between motor motion and processing energy costs. For the model to be used in the experiments, a four-wheeled robot is chosen. It possesses *5 motors:* 1 for each *wheel*, and 1 for the *head*. A single head orientation change is the turn of $\pm 45^o$, whilst it is assumed that a spin of 360^o is enough to relocate the robot to the next grid tile. As such, the cost of the head turn for the motors can be the base cost E_o and the cost for the movement to be $E_m = 4 \cdot 8 \cdot E_o = 32 \cdot E_o$; a notable discrepancy between cost of listening and moving.

Without a frame of reference for the CASA component costs of this implementation in real devices, a limitation of this thesis is the cost that must be assigned to the algorithm. To tackle this issue, pertinent research was explored with results indicating significantly smaller costs for software vs. motor energy consumption [108, 233, 234], even close to values expressed in % or ‰, operating system and algorithm efficiency being the deciding factors. For the purposes of this thesis, and assuming constant operating system functions, more complex software with unoptimised algorithms – or for future implementations using more evolved localisation and tracking algorithms – the cost will be defined as $E_l = \frac{1}{10} E_o$ (i.e. significantly smaller than motor motion costs).

This formulates the total energy consumption function:

$$E_{loss} = a \cdot E_m + b \cdot E_o + c \cdot E_l = E_O \cdot \frac{320 \cdot a + 10 \cdot b + c}{10}$$

a Number of times the agent moved by *1 step.*

b Number of single orientation changes.

c Number of individual listening attempts.

The function plays a role in the final definition related to energy that needs to be presented for the designs of this study – that is the concept of *Performance*. With regards to the tested strategies, performance is evaluated on two fundamental fronts: (a) how much *Time was spent* to reach the target, and (b) how much *Energy was lost*. Obviously, lower values in both indicates better performance for a strategy, whereas high values in either will need to be discussed in the context of the real-world application. A by-product of this twofold classification is that the simulations can suggest strategies to use based on their performance in the corresponding performance indicator of interest. Naturally, failure to locate a target or fully depleting energy reserves fully disqualifies the attempt, given that the ratio of *Successful attempts* (i.e. reaching the target) a strategy has in a set of experiments will also be highlighted in the results. In closing, energy loss and time-to-target will *not* be counted for *unsuccessful* attempts – energy loss is expected $E_{\max}$ and time approaching ∞ (max variable type value in implementation).

4.3 Experiments with Treefrog Behaviours

4.3.1 Introduction

This section introduces the first set of experiments, which has the goal of testing the capability of the original bio-inspired treefrog strategies. Additionally, it provides the first opportunity to fully test the developed framework in a research study. The original treefrog tracking behaviours have been labelled as the *Regular* strategy and the *Explosive* strategy. There term "explosive" here has been inspired by the concept of *Explosive Breeding* (EB), one that is usually applied to seasonal species and especially amphibians, in particular ones that exhibit arboreal locomotion: it refers to the way they swiftly start and conclude a mating season [42, 45]. In the case of treefrogs, this term has also been used in pertinent literature to describe a subgroup of treefrogs in a specific population that have shown much more aggressive tracking characteristics than the rest, hence "explosive" for the corresponding strategy.

The research questions pertaining to the above goals will be presented next, which will drive the evolution of the system. Because these strategies follow typical reactive agent architectures (discussed in an earlier chapter), the implementation is straightforward. The experiment setups are showcased next, explaining the way they can assist in answering the research questions presented earlier. A presentation of the results follows, concluding with a discussion on the findings in respect to the research questions. This research methodology flow will be repeated in each of the following two experiment sets.

4.3.2 Research questions

The purpose of this section is to identify how the two strategies modelled after biological treefrog behaviour perform when attempting to solve the complex problem of reaching a mobile target conserving as much energy as possible. The qualities of the two bio-inspired strategies will be presented in detail next with regards to implementation, but at the abstract level they behave as follows:

Regular Spends more time listening for a target. Moves less and in shorter distances.

Explosive Spends less time listening for a target. Moves more and in longer distances.

The studies already discussed have demonstrated that the treefrogs are efficient trackers with the simplest above strategies. Some of them try to localise their target better and move slower towards them so that they do not lose energy if the target is lost. The rest eschew most of the listening efforts in favour of reaching their target faster, once they believe they have heard one. An interesting finding from biological studies on EB species is that the opposite gender tends to show appreciation to the observation of the physical effort behind it [235, 48]. In contrast, the treefrogs rely on virtually no vision whatsoever towards locating their partner until they are a few feet from each other [42, 48]. Naturally, the effort would be wasted on the partner, and it would not factor in the actual choice of mating with them or not, or potentially waiting for another partner.

One could assume that this explosive strategy is therefore wasted on treefrogs and should have been discarded by natural selection over the course of time. Nevertheless, it appears that this strategy has been retained because it has proven quite effective under certain conditions. Both strategies seem to be opposites, still they serve their purpose well for certain mating environments. EB types of treefrogs tend to be more successful in finding partners during the mating season when there are fewer obstacles in the environment, or there are more potential partners [48, 49]. On the other hand, the regular treefrog breeding behaviour has proven more effective in bio-spaces riddled with more obstacles and less potential partners [42, 235, 48].

These biologically established findings give rise to the question of whether the strategies of the real-world animals can have the same impact when employed by robots attempting to perform a similar task in analogous environments. Meanwhile, it is evident that both strategies have stood the test of time and survived the natural selection process, so they are both necessary for the survival of the species. Still, when the mating season begins, those two types of population – and by extent strategies – find each other competing for the same partners moving around and calling till they find a mate. How would, then, each strategy perform when they compete for the same targets? Therefore, these experiments will attempt to answer questions related to the performance of each strategy under multiple scenarios reflecting such dynamic acoustic scenes.

Combining these curiosities, the research questions for these experiments are formed:

RQ1 How do the strategies perform when tracking a *single* mobile audio source in an environment with *few* obstacles?

RQ2 How do the strategies perform when tracking a *single* mobile audio source in an environment with *many* obstacles?

RQ3 How do the strategies perform when tracking *multiple* mobile audio sources in an environment with *few* obstacles?

RQ4 How do the strategies perform when tracking *multiple* mobile audio sources in an environment with *many* obstacles?

RQ5 How do the strategies perform when tracking *two* mobile audio sources in an environment with *few* obstacles and they are in competition?

RQ6 How do the strategies perform when tracking *two* mobile audio sources in an environment with *many* obstacles and they are in competition?

As a final note, the research questions regarding competing strategies only focus on multiple sources due to what the expected real-world application could be: deploy a few robots to locate as many people as possible in a disaster scenario. The answers from the previous questions are deemed as satisfactory for just a single target. The fascinating – and unknown even in the existing treefrog behaviour literature – outcome, will also become the first emergent behaviour in the system. As stressed before, interaction in a bio-inspired system need not only be in the form of communication for emergence to occur, but rather communication also just bolsters the potential.

4.3.3 System evolution

This section describes the changes required to the system to enable the experiments so that the research questions can be answered to a high degree. Most of the work has already been covered in the previous chapter, which was also the first goal of the project: to create a framework that can effortlessly be parametrised for experiments. The obstacle generation methodology for the acoustic scene via a modified recursive division algorithm has illustrated what *few* and what *many* obstacles would look like. In the meantime, the synthetic treefrog call generation component has demonstrated the capacity to generate multiple calls with different desired characteristics (i.e. pulse rate and frequency). What needs to be confronted is the implementation of the two bio-inspired strategies.

Obstacles Obstacle avoidance is addressed by a simple algorithm: if there was an obstacle towards the direction the perceived target is located at, move away from it and remember that previous obstacle direction for the next *5 steps*. This is an attempt to emulate treefrog tracking memory, which is further assisted by visual perception (e.g. understanding a high bush and moving around it). Initial experiments did not account for this fact due to factoring obstacle detection via an energy-consuming means outside of the scope of this thesis, eventually producing experiment results where the trackers were stuck in corners trying to move towards the target and were stuck in a loop stepping continuously on the same two or three grid tiles. These obstacles are designed as simple movement impediments and not significant enough to factor into sound propagation in the environment.

The first attempt before this type of short-term memory was introduced relied on randomness to control this variable, albeit the improvement was not major particularly in the cases of walled corridors, notwithstanding the fact that it was not analogous to the known treefrog behaviour. One item that needs to be stressed regarding movement is what happens when obstacles are encountered. Much like treefrogs only manage to see their obstacles when they are close, so can the robots in these experiments. It is assumed that some type of extremely low energy proximity sensor is implemented, and the costs do not factor into the energy spent. Energy is primarily concerned with the movement and the listening functions, modelled as described in the corresponding section of this chapter earlier.

Intelligence microservice To this end, the design ought to touch the intelligence or decision-making microservice of the agent. Implementation for these strategies follows the *stop-perceive-act* robotics paradigm reminiscent of the EDBO BioBot life-cycle. The `TrackerHostedWorker` life cycle has already been illustrated in Chapter 3.?, with the focus now being on implementing the `IIntelligenceService.ProcessOwnState` method for each strategy. At this point, both strategies share a common feature – the first step is to check energy reserves and if depleted to notify the `IEnvironmentService` to remove the agent, which ends the simulation if no agents are active.

The agents use the following mixed strategy at this point: am I close to the target, or an obstacle, otherwise do I listen, or do I move, and if so, how far? Consequently, the agent that is to be implemented as a *reactive* model agent has the simple functions based on rules through sensing and acting combinations demonstrated in Figure 4.1. These are translated into simple `if-else` statements in the intelligence microservice implementation, covering the 4th rule depicted in the graph.

Treefrog behaviour showed the following probability numbers for each strategy:

Regular Approx. 60% chance to wait and listen, 40% to move the normal distance (1 step = $1m$).

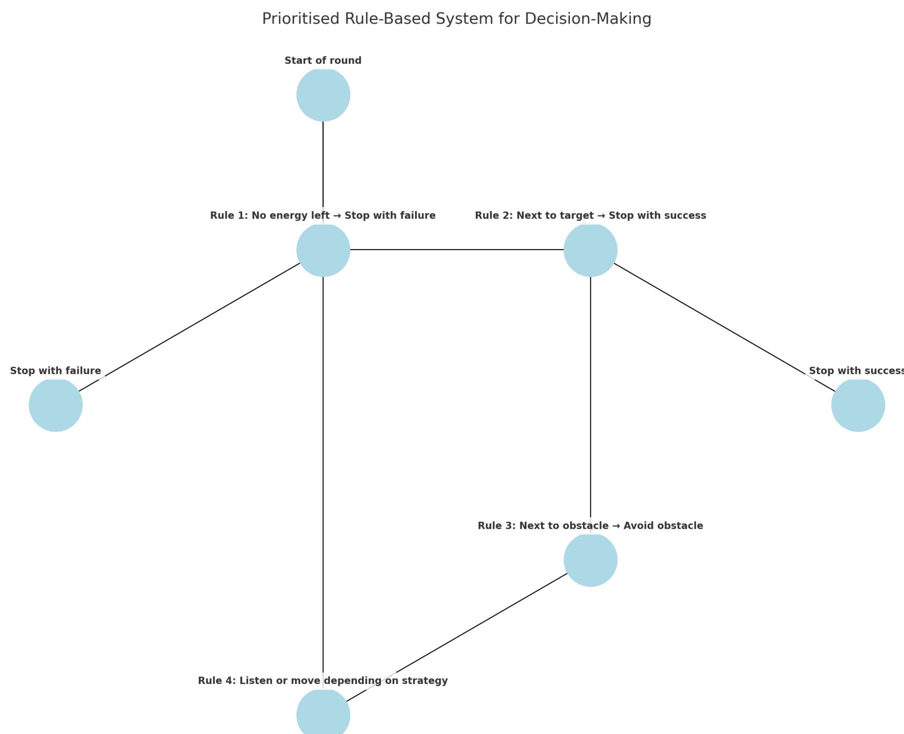

Figure 4.1: Rule-based graph showing how the agent performs in each round of its life-cycle. Rule 4 differentiates between the two bio-inspired strategies.

Explosive Approx. 30% chance to wait and listen, 45% to move the normal distance, 25% to move double the normal distance ($2m$).

Perception microservice The `IPerceptionService` also performs internally the action of rotating the head to solve the front-back problem in ITD computation. However, much like the treefrogs it is modelled after, this occurs once after the very *1st turn* (i.e. *birth* of the agent in the environment), and then only every *5th turn*. Treefrogs, and by extent the tracking robots, do this to conserve energy but also due to movement providing a better frame of reference for continuous localisation of a sound source. Naturally, this front-back algorithm can also be called *out of turn* when the azimuth received by the ITD mapping function is dramatically higher (i.e $\pm 45^o$) than the expect. This entails that the agent, much like real-world treefrogs, expects something closer to the direction they were moving towards before and only acts like this as an exception to ensure nothing went wrong.

4.3.4 Experiment designs

The experiments to be carried out for this part of the study pertain to the evaluation of treefrog-inspired strategies to answer the research questions that emerged. A lot of variables are introduced, with multiple *strategies*, options for *actors* (i.e trackers and targets), and lastly *obstacle* density. The experiments therefore had to be broken down to several categories to ensure that only one independent variable is touched every time. The variables for the experiments will be presented, followed by the setup for each set of experiments.

The **Dependent** variables:

Time The time spent to reach the target.

Energy The energy left after reaching a target.

The **Independent** variables:

Strategy The current strategy followed by the tracker, with values: regular, and explosive.

Targets The number of mobile sound sources in the environment, with values: one, two, and three.

Obstacles The number of obstacles in the environment: low, and high (e.g. Figure 3.4 cases **C** and **D** respectively).

The **Controlled** Variables:

Placement The starting position of tracker, target(s), and obstacles.

Signal The properties of target signals and tracker preferences.

It was discussed previously that when measuring *efficiency* for the strategies the factors are time spent and energy left after reaching a target, if successful, hence they were measured and constitute the *dependent* variables. A lower time spent reaching the target in most scenarios means efficiency on the tracking aspect, while less energy spent means efficiency on the energy management aspect. Since the study focuses on the combination of these two, both were measured.

As far as the controlled variables are concerned, these were variables that are expected to have some effect on the outcome. The relevant position of the tracker to the target can have a big impact in efficiency on both tracking and energy – spawning right next to each other will have dramatically different results than spawning to the opposite corners of the 2D grid, with obstacles in between. With close to $10,000$ starting position combinations and extremely higher numbers factoring in possible obstacle locations, the results would not be viable. This was controlled in the following manner: (a) it was ensured that there is always a path to the target via the designs of the recursive division algorithm and with a different starting column-row for division each time, and (b) the starting positions of both tracker and target were made random. A large number of simulations ($1,000$ for each set) was decided to ensure that a large enough number of possible cases were captured so that meaningful results could be derived.

Regarding the *independent* variables, the best practices are to test only for one variable each time, which was the case for the experiment sets. The variables were chosen for two reasons: (a) there are existing biological studies based on the performance of treefrogs in relating scenarios to compare to, albeit not with hard numbers, and (b) they can cover an assortment of different scenarios a tracker would face in a real-world applications (e.g., tracking targets trapped inside a warehouse floor, tracking targets inside an apartment). Initially, the *regular* strategy and *few* obstacles were constant starting with *one* target and increasing to *two* and *three* in the following sets, gathering results each time. This process was repeated next with *many* instead of *few* obstacles. All the above were then repeated for the *explosive* strategy. In closing, the competitive experiments have only one independent variable, and that was the number of obstacles.

Accordingly, the following were the experiment sets, each with $1,000$ experiments (i.e. simulations):

RF* Regular strategy, few obstacles, * targets.

RM* Regular strategy, few obstacles, * targets.

EF* Explosive strategy, few obstacles, * targets.

EM* Explosive strategy, many obstacles, * targets.

BF2 Regular and explosive (2 trackers), few obstacles, two targets (locate both).

BM2 Regular and explosive (2 trackers), many obstacles, two targets (locate both).

	Success Rate (%)	**Energy Left (%)**	**Time-to-target (s)**
RF1	71	21	129
RF2	72	24	122
RF3	74	22	118
RM1	59	14	141
RM2	58	15	142
RM3	61	18	149
EF1	59	5	52
EF2	65	2	59
EF3	73	8	62
EM1	35	1	90
EM2	49	3	86
EM3	54	3	73
BF2	81	12	94
BM2	79	11	98

Table 4.1: Success rates (finding a target), energy remaining (for successes only), and time-to-target (for successes only) rounded to closest integer for the *bio-inspired* strategies experiments.

4.3.5 Results collection

At the end of each simulation run, a logger implemented in the system stored the dependent and independent variables as a 5-tuple in a plain ASCII format file. The 5-tuple consisted of the values:

$$(T, E, S, O, M)$$

T — *Time* spent (always odd number due to $2s$ *stop-perceive-act* intervals).

E — *Energy* remaining (i.e. $E = E_{max} - E_{loss}$).

S — The runtime *Strategy* of the tracker (binary value *regular* or *explosive*).

O — The *obstacle* density (binary value *few* and *many*).

M — The number of *mobile sound sources* (or targets), an integer between $1 - 3$.

The results were gathered from the files and processed to determine average values for the depended variables of interest, as well as the success ratio, which was derived from where T and E values existed inside the tuple – the code design dictates that these values were not to be logged when energy was depleted. Naturally, this would not provide a metric of how much time it took to deplete the energy, still if it was needed it can be derived based on the implementation of each strategy and the E_{loss} function designed for the model. An overview

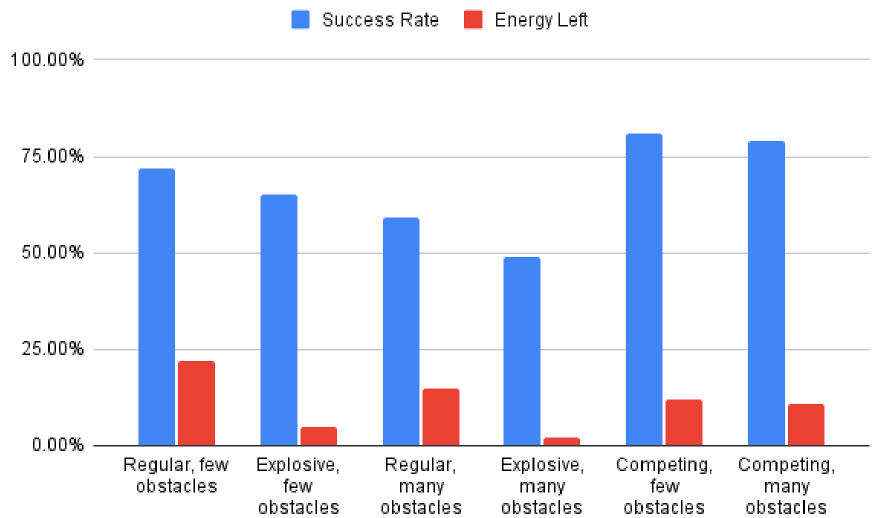

Figure 4.2: A graph representing the metric percentages for the *bio-inspired* strategies in the different environments they were tested in.

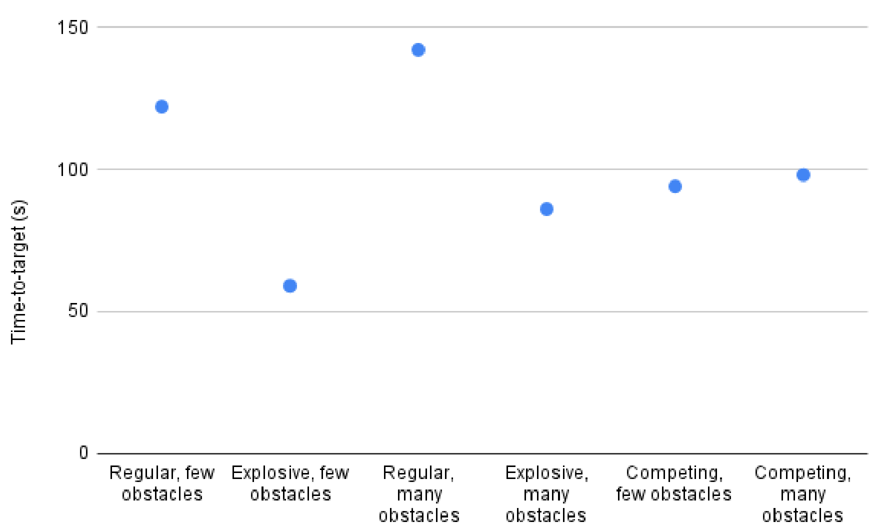

Figure 4.3: A scatter graph representing the time-to-target for the *bio-inspired* strategies in the different environments they were tested in.

of the results, categorised by scenario with regards to the metrics of interest, is presented in Table 4.1 and Figures 4.2 and 4.3. Values S, O, and M are primarily used for the ease of identifying and categorising results.

4.3.6 Regular strategy results

When going over the results of the regular strategy alone (Table 4.1, rows starting with **R**), indeed it verifies what the biological studies suggested about the treefrogs, too: they tend to be highly successful in mating, especially so when there are more obstacles or sources in the environment. While no results with success ratio numbers in similar scenarios exist for the biological studies, the viability of this strategy through its *success rate* of 72.3% across all the individual tracking attempts is clear. Additionally, it appears to retain a similar success rate across all attempts with different number of targets with miniscule deviation, which could also not be attributed to the fact that the tracker attempts to listen for their preferred target more frequently and hence find their way to them better. The larger deviation of approximately −13% compared to the rest was expected when introducing obstacles, however it is close enough to the original to provide a solution most of the time for such cases – and it also is close for the number of targets, too. Overall, in respect to the **success rate** of this strategy, the results are very optimistic.

Going over the results of *energy* remaining after finding the target, it is evident that the strategy shows the capacity for reserving a decent amount of energy for the scenario at hand. Listening and making decisions about the location of the target and next move to make more intelligently, proves indeed that the significantly lower cost of processing power has merits when facing tracking problems. Much like the success rate results, there appears to be once more only a small deviation of energy loss from the increasing number of targets – a 1.3%, which is similar to the deviation demonstrated in the success rate, too. Introducing obstacles does increase energy consumption, but once more the deviation of 7.7% is not that significant and not that far from the success ratio results, proving its consistency. Generally, **energy efficiency** for this strategy appears to be very good and to align with success rate results.

On the other hand, the most significant problems of this strategy are pinpointed when the *time* performance aspect is investigated. For low obstacle scenarios, the same pattern identified above of small deviations to find the target, even with increasing number of targets, can be identified here again. However, the average time of $123s$ for few obstacles does appear to be rather high for the given environment and criteria, especially in comparison with the explosive strategy that will be discussed next. Nonetheless, what is much more interesting is the much higher deviation in time-to-target when more obstacles are brought in, in that it does not follow the pattern identified earlier for energy and success and is disproportionate by $21s$ average (+17). This can be ascribed to the combination of both more listening actions in turn caused by more movement actions to tackle the higher number of obstacles. In closing, **time efficiency** for the regular strategy proves atypical of the strategies energy efficiency and success rate.

4.3.7 Explosive strategy results

The explosive strategy produced more varied and interesting results than the regular strategy in general (Table 4.1, rows starting with **E**). To begin with, the analogous biological studies alluded to a higher success rate in comparison to the regular approach under specific conditions: more targets and less obstacles in the environment. The results for this strategy did validate the part of high success rate expectations under such conditions, nevertheless they did not achieve the much higher success rate expected of them as compared to the regular strategy (difference of -9.9% instead of gains). This could be accredited to the design of the experiments, and specifically to the values of E_m as compared to E_{max}. An outlier to the success rate results, something unexpected was the performance of the strategy with many obstacles (where it is expected to be lacking) but with also several targets (max of three), with results deviating merely -4% from the regular strategy. These results were explored with added tracking to the logger of the run, to determine why the case was so: the agent managed to reach one of the three targets, even if they weren't the most preferred one given by their randomly assigned signal preferences. Overall, **success rate** for this strategy was not acceptable except for very specific conditions, raising concerns for its general viability for dynamic scenarios.

Energy efficiency for this strategy was also found to be lacking. Success rate is defined by the ability to locate and reach the target before going out of battery. In the case of the explosive strategy, looking at the results for success rated discussed above it is crystal clear that most of the time the virtual robot ran out of energy before it was able due to the higher emphasis on choosing movement, and moving more tiles per turn, to listening instead. A specific simulation could have been the following: target could have moved the very next turn after moving 2 tiles, and follow again with the same action choice, where a target has moved too and repositioned themselves inside a room and towards the exact opposite direction. Even for the successful attempts, the pattern followed demonstrates that the energy was barely sufficient to be successful in the cases with obstacles (average of 2%), whereas the results were far more encouraging (difference of $+3\%$) in the experiments with a lower number of obstacles. Generally, **energy efficiency** was expected to be lower than the regular strategy, but not so much that the success rate in the scenario could be affected as greatly.

Finally, time-to-target for the explosive was analysed. This is where the strategy performs extremely well compared to its competition, when viewed outside of the overall context of a real-world application or success itself. For the far less times it did manage to complete the objective, it did so almost 3 times faster than the regular strategy when there are fewer obstacles, and reaching up to approximately 2 times faster with more obstacles introduced to the environment. These results also do align with the energy consumption results and highlight the "explosive" nature of the strategy. In conclusion, the

time efficiency of this strategy is exceptional and something that could be exploited under specific conditions.

4.3.8 Competing strategies results

The two strategies were pitted against each other in a very specific scenario: attempt to find two moving targets in the environment, whether there are few or many obstacles. This choice was made primarily due to the scenario providing the middle ground, but also to determine how the two strategies fared when another tracker was also active in the operational space. The simulation completion parameters were changed as compared to the previous simulations, merely to ensure that the simulation ends again when at least one target is found, or both trackers lose all their energy, it turned into: find both targets, or both trackers ran out of energy.

The logger gathered the same 5-tuple of (T, E, S, O, M) ,but this time for each tracker (Table 4.1, rows starting with **B**). As communication among agents was still not implemented, any interaction they had was from the positioning of each other, but also from the success of each other. Even this low-impact, low-level interaction, nonetheless, can produce emergent phenomena for the system. The core difference with this scenario is that it is regarded as a cooperative problem, therefore the results are dissimilar to the individual ones before as they consider the success rate of the scenario, and the time or energy efficiency of both actors, thereby of the overall "deployed" system itself.

With regards to **success rate**, the results for the overall performance were dramatic. Overall success rate for the scenario of locating at least one target was increased far beyond both individual strategies (reaching an average of 80%), which is welcome in real-world applications such a system could be used for, and an +7.5% average increase over the best so far (the *regular* alone). These observations present the first emergent phenomenon in the system, where the interaction of the agents in the microscopic level (i.e. the *explosive* strategy stealing rewards from the *regular* strategy) created a unique interaction, which in turn produced desirable in fact results in the microscopic level (i.e. increased overall system success rate).

Going over the **energy efficiency** specifics, the result showcase that the *explosive* strategy once more seemed to have a larger impact in the result, with high costs incurred towards achieving the goal due to explosive-type movement. However, a problem in the system was outlined: the total energy loss is much higher than selecting the best one for each scenario *regular* for both scenarios, and somewhat better than selective *explosive* for the one where it seems to perform best. What was ultimately observed is that right after the *explosive* tracker ran out of energy the regular *tracker* could keep going, thereby most of the times being successful even if at the much higher cost of energy, in importance trade-off.

Finally, **time efficiency** also follows the previously observed pattern: the *explosive* strategy brought its unique traits once more and improved overall efficiency of the deployed system when successful, with the *regular* one picking up its pieces as needed to increase system success rate even at the extra cost in time. This was true for either obstacle density scenario, too. However, another important observation is that the time-to-target for the scenario as compared to the results of the *regular* has been decreased by an outstanding $28s$ and $44s$ for few and many obstacles respectively, a by-product that is estimated to have been caused by its obstacle avoidance troubleshooting relying on the other tracker, especially so in the case where they happen to start tracking the same target.

4.3.9 Discussion

To derive these results for a better overview of the strategies and how they compare to each other to answer the questions, some elementary metrics have been looked at utilising existing provided values and functions. In general, the *explosive* strategy demonstrates an average success ratio, notwithstanding it is significantly disproportional to the energy loss over time as compared to the regular strategy. It excels in the tracking of *at least one* target when it is presented with many possible targets and especially in a less-obstructed environment, still proving useful even when there are more obstructions in the environment with many targets to track.

On the other hand, the regular *strategy* has an exceptional overall energy loss over time that tends to result in a much higher success rate for every scenario. However, this is where the second most importance divergence in the metrics plays its part from the time-to-target analysis. The significance of the far shorter time required when using an explosive strategy, at the cost of the energy lost notwithstanding, can be significant depending on the real-world application of the energy-efficient mobile sound source tracking. Indeed, if there is a disaster scenario where every second counts towards finding a target, the choice could be to rely on the *explosive* one, whereas in a scenario where leniency in time is allowed then the *regular* approach is preferable for higher accuracy.

Consequently, regarding research questions **RQ1**, **RQ2**, **RQ3**, and **RQ4**, barring the ones related to competition, the answers can be summarised as follows: the *regular* strategy can be anywhere from ??% to ??% more successful in reaching the target, at an energy cost that wanders between twice or four times less than the *explosive* strategy, although at virtually the same cost in time proportionally. As far as the competition between the strategies is concerned for **RQ5** and **RQ6**, the results highlight the fact that the *explosive* strategy is the winner for most cases, since it tends to reach the targets faster than the *regular* one – bot only when it is capable of localising them properly. Nevertheless, the two appear to complete each other offering increased success ratio under the specific scenario.

Both time and energy have been designated as the factors that define the level of efficiency of each strategy, yet it is becoming apparent that the two bio-inspired treefrog strategies might be related to a very-basic trade-off here based on the conservation of energy: *regular* prefers to put the energy to use in a much more meaningful manner expecting that the outcome will be achieved, whereas the *explosive* attempts to solve the problem as fast as possible with a view to minimising lost energy (pertinent the arboreal locomotion behaviour). The majority of treefrog populations have evolved to survive the problematic mating seasons in hostile environments following this approach, and with the competitive scenario requiring to locate both targets the emergent phenomenon of *complementarity* was identified [236]. Indeed, the two strategies complement each other in locating both, with the *regular* strategy making up for the failures of the *explosive* one.

The complementarity implies two things: (a) there is merit in running experiments with more complex scenarios to determine how many trackers of each strategy could solve a more complex problem together to reduce overall system energy requirements, and (b) there is reason to investigate whether a combination of the two strategies might be capable of solving the problem more efficiently on its own. While (a) would be an interesting avenue to explore, the focus of the study is to develop first and foremost a better strategy for the single agent, an adaptive one that can overcome more efficiently the problem, therefore constituting (b) the best option of the two. Introducing better awareness than the typical reactive approach the bio-inspired strategies employed could result in more meaningful choices between acting and perceiving that would provide a more efficient and effective solution.

Meanwhile, some limitations and weakness of the experiment and system design for the simulations have also emerged through the experiments. The core problem refers to the assumptions on the energy availability and costs for the actions performed by the intelligent agents acting now as virtual robots. As detailed in energy management discussion before with regards to energy, care was taking through pertinent literature to derive a proper analogy of processing costs to motor costs, albeit each different hardware model could have different results.

In conclusion, verification of the experiments could be performed at more rigorous level by having hardware availability that can run the system. Then the hardware could be left running the operating system with listening algorithm on loop to derive proper values for everything, as well as running motors on a separate experiment to determine proper energy – and eventually compare it to the max value of its battery. While costs have been cared for, total energy available to the virtual robot was assumed to be low enough so that the strengths and weaknesses of each strategy can be highlighted. This decision will remain stable throughout this thesis, to have meaningful results to compare to with the upcoming optimisation attempts. **Objective A2-O1** has been achieved at this point.

4.4 Evaluating a Combined Approach

4.4.1 Introduction

The bio-inspired strategies that were explored relied heavily on the observed behaviours of frogs, both in respect to their energy management as well as basic perception and knowledge of the world, such as the limited memory for obstacle avoidance or more simplistic decisions regarding choosing the proper angle out of a few they could localise and identify as relevant. This imposed some limitations to the performance expected of a modern system that could use more complex capabilities and processing tasks to determine the best choice. It was presented that the literature on agents suggests that their autonomy and efficiency in performing their highly specialised tasks naturally stems from having a better view of the local system than the whole. Therefore, the first step for a better strategy could be letting the agent know more about its environment, obviously limited only to its available perception system but including more knowledge about its previous actions, too.

The *complementarity* of the two strategies also highlighted a new path towards contriving a better strategy. Each of the two strategies were shown to have its strengths and weakness, so the question was raised of how an agent could perform when attempting to solve the problem when they could switch between each strategy at will. Treefrogs are more one-dimensional in the strategy the follow, and if the mating season ends without having found a partner, they will try again in the next season following the same approach. An intelligent agent, however, can make more informed decisions, especially with an expanded knowledge of the state of the world through their previous actions and perception attempts.

The fundamental concept here is that the agent can make an informed decision on how to act. RL has a similar approach, where the agent has some knowledge on the state of the world and makes more informed decisions after the training through the reward function, and chooses the right policy (i.e. *regular* or *explosive*) to apply for actions. While this will be explored at a later step, the core idea behind the RL decision-making process will be adopted for this *combined* strategy. To this end, the agent should wake up and evaluate the state of the environment as it knows it and then make a decision between moving or listening based on the strategy that they believe would be the best under the circumstances.

A path to problem-solving combining the best of each approach now that it has been determined through the experiments could bolster overall strategy success and produce one that is better at dealing with even more dynamic environments than simply moving targets. This could be new sound sources entering the scene, or more obstacles appearing in the scene, and the agent understanding that the *explosive* would be a best solution to the former or switching to the *regular* for the later. Accordingly, this section focuses on the design of the system to establish this strategy, the research questions that need be answered and the experiments design to this end. The results will be present and discussed, looking into the performance of this strategy and what can be capitalised on from its design for an even more elaborate adaptive strategy.

4.4.2 Research questions

The purpose of this experimental iteration in the study is to determine the capability of the new *combined* strategy to address the energy-efficient tracking of mobile sound sources. With the results of the previous experiments now providing a substantial foundation to compare performance against (given that the biological studies provided no such hard data), naturally the first set of research questions of interest revolve around the performance of the combined strategy under the same scenarios. This includes tracking one or many targets in an environment with a few or many obstacles. Additionally, competing agents with the combined strategy will also be used to try and find both targets, looking for performance in such cases, too. The questions are thus analogous to the previous **RQ1-RQ4**, and **RQ5-RQ6** respectively. Meanwhile, a new question has been formed through the speculation of the expected capabilities of the new strategy in more complex environments, and by consequence problems.

Accordingly, the following questions need to be answered:

RQ1 How does the combined strategy perform when tracking a *single* mobile audio source in an environment with *few* obstacles?

RQ2 How does the combined strategy perform when tracking a *single* mobile audio source in an environment with *many* obstacles?

RQ3 How does the combined strategy perform when tracking *multiple* mobile audio sources in an environment with *few* obstacles?

RQ4 How does the combined strategy perform when tracking *multiple* mobile audio sources in an environment with *many* obstacles?

RQ5 How does the combined strategy perform when tracking *two* mobile audio sources in an environment with *few* obstacles and they are in competition?

RQ6 How does the combined strategy perform when tracking *two* mobile audio sources in an environment with *many* obstacles and they are in competition?

RQ7 How do the strategies perform in a *highly dynamic* environment?

A *highly dynamic* environment for the purposes of this thesis is defined as an environment where the defining parameters do not remain static but may change over time. The defining parameters in the scenarios generated for this thesis are the number of actors and the number of obstacles. Therefore, in such an environment new mobile sound sources will be introduced over time and new obstacles can be added, too, while the agent attempts to track their original target and overcome the new obstacles. Answers on what the viability of the new combined strategy is in a highly dynamic environment are crucial to determining how capable the strategy is at switching from one strategy to the other on the fly to be more efficient with the new state of the world. To have a comparison basis, experiments need to be run for the other strategies, too.

4.4.3 System evolution

Implementation of the *combined* strategy involved studying both implemented bio-inspired strategies and taking under consideration the answers to the previous research questions in this chapter. These answers were pivotal in determining when the agent will decide to adopt which strategy for its problem-solving capabilities. However, to be able to decide the agent will need to become more intelligent, which primarily entails retaining a better state of the world so that decisions can be made. The key elements that comprise the environment, as well as the problems the agents have encountered in past attempts, factored in the decision of what must be remembered and considered towards decision-making.

This section describes the work towards materialising this new, artificial strategy inspired from the *complementarity* the bio-inspired strategies exhibited. The design process at this point also considers possible future needs for the adaptive strategy, and by extent for laying the foundations for enabling RL implementations. To this end, the architectural concepts of a state space and action space are introduced. Furthermore, policies are also introduced at the same time, so that the agent can choose the best policy to follow based on the states. The reward concept was not of consequence at the time, due to the lack of a learning process as well as a proper design for rewarding the agents at this stage. However, the state and action spaces in combination with the policies were enough to design an aspiring, artificial strategy.

4.4.4 Action space design

The action space for the implementation of this strategy was the more straightforward to create. This is attributed to the fact that the only requirement is to create a set of all the actions an agent can perform in any strategy and bring them together:

- Move one tile.
  - Implemented for 8 orientations.
- Move two tiles.
  - Implemented for 8 orientations.
- Attempt to localise a target.
  - Additionally solve the front-back problem.

The above actions are all that an agent can do, albeit the agents take these actions with a randomised approach at each interval with the bio-inspired strategies.

Having only 2 tiles as an extra move action for the explosive strategy is essential to observe the impact of such a minor change to the system while staying true to the bio-inspired strategies. A potential higher number of movement tiles possible could have far more drastic effects and should be studied separately in experiments, potentially highly expanding the scope of this study. For the *combined* strategy the policy dictates which action will be chosen from the action space. For any future RL implementations, this state space is expected to remain largely unchanged.

4.4.5 State space design

The agent was infused with a new `State` with more properties to describe the environment. In the past, the agents were only aware through the state only the direction of the last encountered obstacles for memory and obstacle avoidance properties. This was the starting point in trying to determine what would be of interest to an agent pertaining the state of the environment, although it should be something they *can* know. Overview of agent capabilities based on its perception systems, both the listening and the obstacle detection systems (i.e. a proximity sensor), produced the following list:

- The number of signal peaks identified (i.e. sound sources of interest).
- The cues of importance (i.e. base frequency) for each sound source that was identified.
- The angle at which a specific source with a particular signal was identified.
- The direction at which obstacles were identified around.

These capabilities of the agent can be combined to develop a more advanced memory and perception mechanism. The two bio-inspired strategies have been designed to only keep some of that information only and not utilise them beyond choosing a target closer to the preferences of the tracker, as well as knowing where obstacles were encountered recently to avoid going that way. These were chosen as they simulated the basic observed behaviours of the treefrogs, and so they did not incorporate memory or knowledge about number of possible targets. Still, the artificial strategy need not be constrained by such bounds, therefore the following intelligent functions memory functions pertaining to the state of the world were added:

- An ordered list of the angles each target appeared to be at for the last rounds relative to current orientation.
- A number representing the possible targets to track.
- An association of the above with the preference function for selecting a target.
- An estimation of the number of obstacles encountered in the environment.

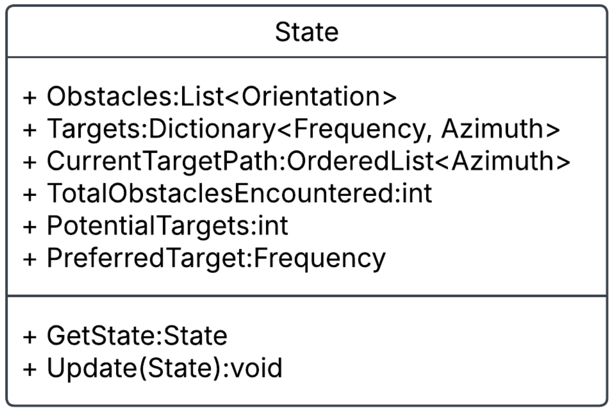

Figure 4.4: UML diagram depicting a `State` class that defines the State space items.

The state space, in conclusion, for the purposes of this approach that does not reflect a true RL implementation, is constraint only to the next states the agent can be in. It is expanded from the original with the above properties and can be used by a policy to determine the best course of action. Figure 4.4 shows an abstract depiction of the class used in the system for describing States in the State space, illustrated in the Unified Modelling Language (UML).

4.4.6 Policy design - new intelligence microservice

The above memory functions can serve as a catalyst to the agent for deciding on which policy to follow, which means when to switch to a *regular* approach or to an *explosive* one. The `IIntelligenceService` was hence redesigned to make the decision through a new `IPolicyService` that receives as input the state of the world returned by the `IPerceptionService`. The policy was designed to return a proper action from the action space described next based on the state, much like the policies do in police-based RL, although without the function mapping between states and rewards that did not exist.

The checks that are performed, and have been derived from the answers to the earlier research questions, are as follows:

- Number of sound sources count weighs more heavily than obstacle density count (66% – 33% respectively).
- When there are few targets, try to take *regular* strategy actions.
  - Otherwise try to take *explosive* strategy actions to better reach any.

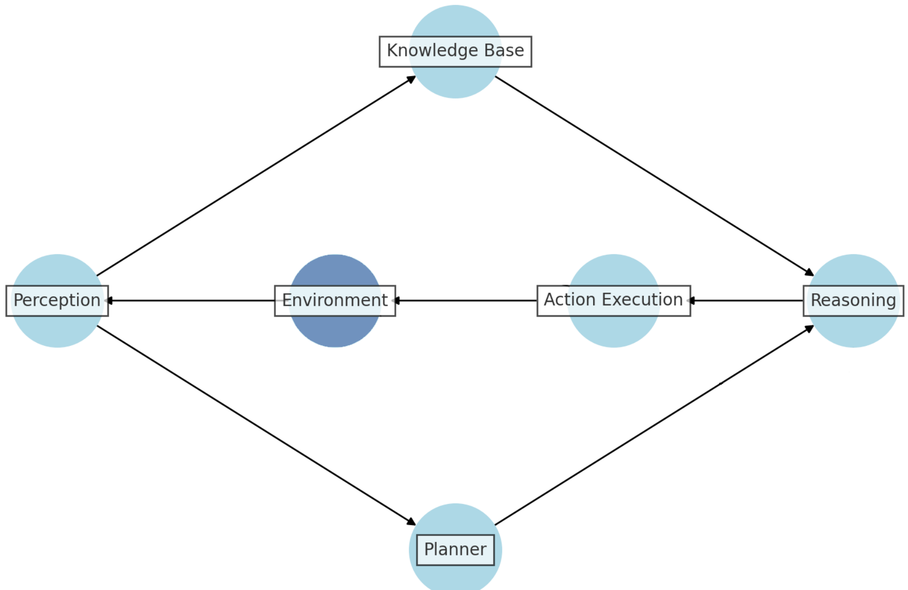

Figure 4.5: *Deliberative* agent architecture – core components and interactions among them. *Environment* is the external component, all else belongs to the agent.

- When it is estimated there are few obstacles, try to take *explosive* strategy actions.
  - Otherwise try to take *regular* strategy actions to better avoid them.
- If the target seems to be moving a lot (i.e. angles changing by more than $\pm 30^o$), check number of targets:
  - If only a few and the other target has large deviation from preference, try to take *regular* actions.
    - Otherwise switch to *explosive* actions to cover more ground.

This covers the basis for the policy directing which type of actions to take. When the algorithm above mentions to "try and take actions from another strategy", it refers to the fact that the strategy swapping will not be fast. To elaborate, the agent will try to take at least 3 turns in a strategy before re-considering whether a better approach will be needed.

A new implementation for the `IIntelligenceService` interface has been created that can be used for the combined strategy. The proposed model for this type of agent has been the *Deliberative* model – a model close to the previously used *Reactive* one but with the extra layer that handles the more advanced decision-making. The general architecture of Deliberative agents is depicted in Figure 4.5.

The components showcased there are:

Perception Gathers data from the environment.

Knowledge Base Stores information about the world.

Reasoning Evaluates and makes decisions.

Planner Determines the best course of action.

Action Execution Carries out actions in the environment.

Environment The external world with which the agent interacts.

How this architecture translates into the *Combined* strategy agents:

Perception The `IPerceptionService` of the agent (with the core function of listening).

Knowledge Base The *State Space* design described above.

Reasoning The high-level decision making in the *Policy* design described in this section.

Planner Either the *Regular* or the *Explosive* strategy as chosen by the Reasoning component above.

Action Execution Would be the *IActionService* of the agent.

Environment Real-world or simulated environment (`IEnvironmentService`).

This is part of the reason why Action and State spaces, as well as the Policy design that ties them together, were developed at this point during the study, the other part being preparation in the system for the true RL implementation to follow. Essentially, the new implementation titled `CombinedStrategyService` is a mapping of the rule-based approach above using the new `IPolicyService` and serves as the deliberative layer of the intelligent agent. As per the model requirements, the State space described above serves as the internal world model that is needed for the deliberative reasoning layer.

4.4.7 Experiment designs

The experiment designs follow the patterns outlined in the previous section, given that most of the experiments will be repeated to answer both the previous and the new research questions, but this time only with the *combined* strategy in focus. The variables are also the same, although this time the *Strategy* variable will not be changing across experiments except for the last one for which no prior results exist.

This shapes the experiments into the following list:

	Success Rate (%)	Energy Left (%)	Time-to-target (s)
CF1	69	17	83
CF2	75	18	85
CF3	79	14	91
CM1	58	13	92
CM2	63	14	96
CM3	69	12	99
DF2	82	7	95
DM2	83	8	97

Table 4.2: Success rates, energy remaining, and time-to-target rounded to closest integer for the *combined* strategy experiments.

CF* Combined strategy, few obstacles, * targets.

CM* Combined strategy, many obstacles, * targets.

DF2 Dual combined trackers, few obstacles, 2 targets (locate both).

DM2 Dual combined trackers, many obstacles, 2 targets (locate both).

HDR Highly dynamic environment, one *regular* tracker.

HDE Highly dynamic environment, one *explosive* tracker.

HDC Highly dynamic environment, one *combined* tracker.

Except for the experiments of the highly dynamic family (**HD***), the designs for the rest of the experiments are already familiar. In this case, a high degree of randomness is involved. Targets (except for the one of interest to the tracker) enter at random intervals reaching up to 3 sound sources in the environment (average of 20s intervals). After a random and medium amount of time (average of 50s) one target that is not of interest leaves, specifically the 2nd target that enters to confuse the tracker. Obstacles are also increased over time, which is a jump from the category of *few* to *many* as seen in past experiments (the maze generation algorithm runs with new settings and updates the environment) but stay till the end of the experiment runtime at the new *high* state. This happens during an average of 60s after the start of the simulation. The design of this experiment attempts to confuse the tracker by creating the more dynamic environment, to determine how each strategy can tackle such an advanced problem (details in Appendix A).

4.4.8 Results

An overview of the important results with the pertinent metrics are presented in Table 4.2, which can be compared to Table 4.1 to determine any significant changes that have emerged with the new strategy. The (T, E, S, O, M) 5-tuple is used once more for logging and analysis purposes with the S remaining constants for all experiments in this batch. In advance of the summary for each individual aspect of interest (i.e. success rate, energy efficiency, time efficiency), the results showed that the *combined* approach manage to improve most metrics across the board, some significantly and some not. First, the results pertaining to the similar past experiments will be discussed, followed by a separate analysis for the **D*2** and **HD*** experiment sets.

With regards to the **success rate**, the results appear to be close to the results for the *regular* strategy when facing few obstacles: +3% to +5% better for the cases with multiple targets and few obstacles, although −2% worse in the single target scenario. These deviations were slightly more exaggerated with the introduction of more obstacles (+5%, +8%, and −3% respectively). The extra costs introduced by virtue of borrowing behaviours from the explosive strategy may have contributed to the reductions in the few cases where obstacles could not be escaped, whereas the increased movement ought to have helped when tracking multiple targets as it did with the pertinent scenarios in the *explosive* strategy results.

The **energy efficiency** results were, unfortunately, not as clear to determine despite being similar across all cases. Indeed, they do not appear to follow closely for specific families of problems either of the two strategies examined before. The cause for this can be the swaps between strategies being not so consistent, or at least not as expected. To clarify, there could be cases where the agent tends to listen a lot in the *regular* strategy and then switching to the *explosive*, only to fall to the smaller chances that they listen again, thus ending up with a higher energy result. The same, evidently, could have happened with movement, producing such difficult to make sense of results with regards to efficiency. One clear property that could be identified through these experiments was that the overall E_{loss} in the system was increased, naturally due to the increased movement introduced and thus not getting benefits from the *regular* strategy part of the combination.

In contrast to energy, **time efficiency** delivered the most important results for this batch of experiments. Across all experiments, the time was greatly reduced with relation to that of *regular* attempts at solving the problem; in fact the difference between the two strategies was almost cut in half (range of $37 - 46s$). In tandem with the success rate being close to that of the *regular* strategy, this would suggest that the *combined* strategy could effectively replace the *regular* strategy due to the sheer decrease in T in spite of the volatility of E_{loss}.

The observations from the experiments related to the competition of the trackers for two targets (**D*2**) did not provide any significant results. The results only have a small average deviation of +2% to success from the existing ones for the two competing strategies and thus there are no observations of note, other than a small increase in E_{loss} in the overall deployed system (−4% average less energy left). At the same time, no emergent phenomena were observed through the interactions of the similarly behaving agents. Where the two strategies appeared to complete each other if seen as a CPS instead of competition, this strategy that borrows the best behaviours from both seems to

	Success Rate (%)	**Energy Left** (%)	**Time-to-target** (s)
HDR	44	9	152
HDE	13	2	101
HDC	66	7	109

Table 4.3: Success rates, energy remaining, and time-to-target rounded to closest integer for *bio-inspired* and the *combined* strategy.

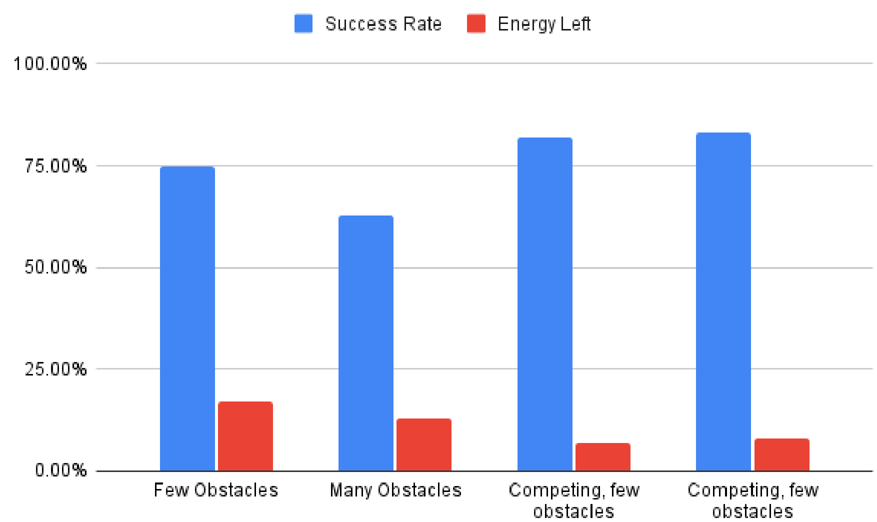

Figure 4.6: A graph representing the metric percentages for the *combined* strategy in the different environments it was tested in.

have the capability to solve the problem in the same manner but does not bring forth any noteworthy observations that could be used for improving strategies in the future.

Lastly, there was the **HD*** collection of experiments, still a new category that did not have prior results to compare to. The three strategies were put through the test and the results demonstrate the difficulty of this scenario (Table 4.3). Success rate plummeted for both bio-inspired strategies, especially so for the *explosive* strategy hence establishing it as a prohibitive choice for such problems, while at the same time the *combined* strategy appears to perform almost 50% better than the *regular* one (achieving the goal 2 out of 3 times). Regular *energy* efficiency reached *explosive* levels, too. *Combined* strategy still retained an impressive time efficiency (roughly $^2/_3$) as compared to its only real competitor for this scenario, thankfully at a very slightly lower energy efficiency (-2%).

Overall, the most important result with regards to combined is that the

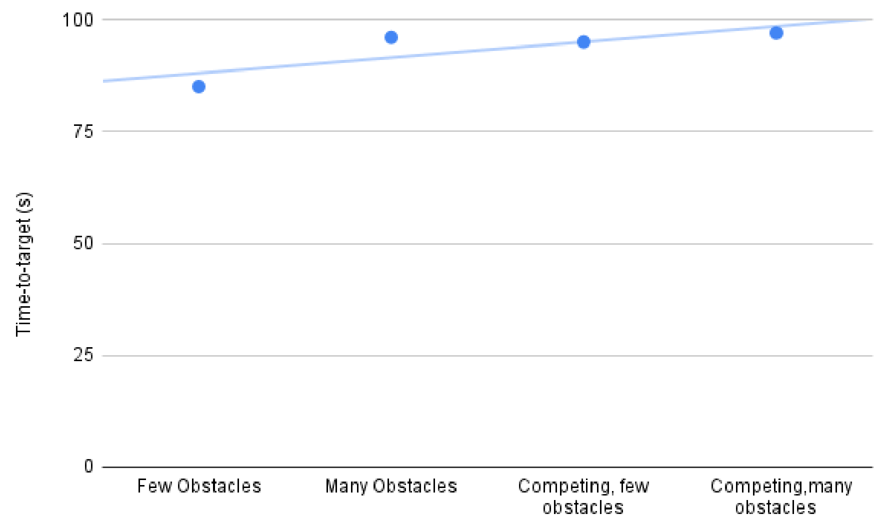

Figure 4.7: A scatter graph representing the time-to-target for the strategies in the different environments it was tested in. Trend line indicates the rising difficulty of the different scenarios.

metrics across all cases have been brought up to a very uniform and hence predictable number despite obstacles or target numbers, which could comprise one of the strengths of this strategy – stable and thus predictable performance across most environments as illustrated in Figures 4.6 (success and energy rates) and 4.7 (point closeness to trend line indicating predictability in time-to-target).

4.4.9 Discussion

The analysis of the observations for the *combined* strategy with respect to the research questions posed could be pinpointed to a single fact: this strategy has success rate akin to that of the regular strategy but reaches its goal in significantly shorter time at an energy cost far lower than the time gains. Indeed, this strategy does achieve a better balance of the two strategies with a sufficient efficiency between time spent and energy consumed, establishing it as the best choice for the three in most case, with an apparent *predictability* across most cases with regards to all metrics. The best traits of both strategies were thus combined successfully, and it is evidenced by the results, while the implemented logic of switching between strategies using the newly expanded local view of the world state the agent possesses has been a great tool in achieving this outcome. This is an answer that holds true for all the research questions **RQ1-RQ4**.

On the other hand, the answer to questions **RQ5-RQ6** is not encouraging. No significant improvements were observed when the combined strategy is used on multiple agents for a CPS attempt that **DF2** and **DM2** encourage. In fact, the utilisation of the two bio-inspired strategies at the same time does seem to offer a slight increase in energy efficiency. The reason why this happens is not readily apparent through the gathered metrics, which implies that maybe a more thorough investigation for this scenario and the *combined* strategy in CPS is required. Investigation of strategy swapping times did reveal that the small amount of randomisation the new algorithm adopts provides more predictable behaviour in a less dynamic environment, which connotes that a more advanced and adaptive strategy would require a more advanced decision-making algorithm – or a higher degree of randomisation.

Meanwhile, the newly introduced concept of a highly dynamic environment and an experiment design for it delivered another basis for testing the capabilities of strategies. The goal for an adaptive strategy is to offer the best results under any circumstances by being able to make the best decision at any given moment no matter the state of the environment. Accordingly, a scenario like that is ideal for determining how the strategies can react, both how how fast and how efficiently, and eventually manage to solve the problem. **RQ7** was centred on this scientific need pertaining to the developed strategies, and it has been answered, too: *explosive* is almost entirely incompetent, while the much-increased energy over time efficiency of the *combined* strategy establishes it as the winner. Nonetheless, these results are not encouraging as compared to much better performance in more specific scenarios for some of these strategies.

Moving forward, a question emerged earlier with regards to how to best steer these attempts towards a more adaptive strategy: more informed decision-making, or more reliance on randomisation with the *combined* approach. While this last solution (i.e. randomisation in decision-making) has been debated over the years, and there have been historical cases where randomisation has improved decision-making in lieu of the more deterministic nature of a strategy, the new tools the field of AI offers can be exploited to formulate more adaptive strategies instead. Having studied the different machine-learning and AI paradigms, there is opportunity in leveraging the existing developed strategies and tools (e.g. world state and CASA) developed throughout this thesis via the means of a more robust AI than more into randomisation sampling and related techniques for adaptivity in highly dynamic environments. This concludes the work towards realising **Objective A2-O2** of this thesis.

4.5 Towards an Adaptive Strategy

4.5.1 Introduction

RL has basked in the recent years in significant advancements, both in practice and in theory, and is well-known for its applications in an assortment of different and interdisciplinary domains, including but not limited to robotics, healthcare, finance, gaming, and education [177, 179, 174]. When looking into advanced machine-learning techniques that could be used to improve upon the bio-inspired and other artificial approaches to solving the problem this thesis revolves around, RL offers an array of unique properties that can be beneficial to this end. Most important of which, however, is that it is an active learning approach where the agents continuously improve by interacting with the environment. This is a case of real-time learning as opposed to static training datasets that the other core machine-learning paradigms employ.

Summarising the findings in the researched field earlier, RL follows an unsupervised approach, therefore no training specific sets infeasible to create for such scenarios are needed, an additional boon for development purposes. Additionally, RL need not always rely on an explicit modelling of the environment, which can range from inaccurate when it is dynamic to infeasible depending on the problem and the variables involved, therefore it can be argued that a model-free RL approach is robust to uncertainty. Meanwhile, the policies developed can also ensure that the best actions are taken based on the new state of the world, one more tool in the arsenal against this complex problem. Finally, the path planning involved in this case has been found to be a strength of RL for complex scenes, due to the capacity for adapting to random starting and final positions by finding the best path forward. Most importantly, though, it is the fact that the core concept of maximising the eventual rewards, which in this case could be the remaining energy.

With the overwhelming strengths of RL for this scenario, the focus then shifts towards what the best RL algorithm is to employ. While recent advancements in Deep Reinforcement Learning applied to the traditional approaches has produced outstanding results for several applications [179, 177], designing and developing the sheer amount of data needed for the training on this dynamic acoustic scene defeats one of the reasons why RL is chosen. Accordingly, more traditional algorithms ought to be considered that can be implemented within this step and the scope of this thesis. There are several properties and parameters that affect the type of approach as related to the problem, because these define an algorithms strengths and weaknesses. A preliminary review of these was presented in the background theory discussion for this chapter, therefore outlined next is the reason why a Q-learning, ε-greedy approach has been chosen.

4.5.2 Research questions

Having entered the third step of an iteration on strategy making and evaluation in the search for a proper adaptive strategy, the previous research questions **RQ1-RQ7** remained relevant for this step, too. Question **RQ7** is the most important question to answer, specifically because the *highly dynamic* environment is the one that demands the highest degree of adaptivity. The principal reason for developing the new *adaptive* strategy was to reach a point where the overall energy-efficient tracking solution via CASA would be capable of adapting to any situation at any time. Consequently, the reinforcement learning approach followed here will also be attempting to reach that state and this can be evaluated by repeating the experiments with the new strategy.

Furthermore, one more question emerged through the consideration of the best strategy to use to this end: is the chosen approach of a Q-learning, ε-greedy strategy appropriate for solving this problem? Several other approaches and strategies have been considered, and this one was chosen for its identified properties via research and pertinent literature. Nonetheless, there are some problems with this approach that will be discussed in relation with the answer to this question when the experiments are concluded. Moreover, this new question aspires to determine whether this RL approach is viable for future studies by researchers in the field by more in-depth optimisation studies, or if such research should probably lean towards a more optimal solution.

To conclude, the research questions for this iteration of experimentation are:

RQ1 How does the adaptive strategy perform when tracking a *single* mobile audio source in an environment with *few* obstacles?

RQ2 How does the adaptive strategy perform when tracking a *single* mobile audio source in an environment with *many* obstacles?

RQ3 How does the adaptive strategy perform when tracking *multiple* mobile audio sources in an environment with *few* obstacles?

RQ4 How does the adaptive strategy perform when tracking *multiple* mobile audio sources in an environment with *many* obstacles?

RQ5 How does the adaptive strategy perform when tracking *two* mobile audio sources in an environment with *few* obstacles and they are in competition?

RQ6 How does the adaptive strategy perform when tracking *two* mobile audio sources in an environment with *many* obstacles and they are in competition?

RQ7 How does the adaptive strategy perform in a *highly dynamic* environment?

RQ8 Is Q-learning suitable for solving the energy-efficient CASA tracking problem?

4.5.3 Implementing Q-learning for CASA

One challenge for the implementation of Q-learning as a strategy in the current system was also an opportunity to evaluate its architectural design. The initial architecture was designed to facilitate the previous strategy implementations, however the demands of the system in implementation given the new machine-learning requirements were much higher. The ought to be able to leverage its designs for supporting both RL training and Q-learning operations with the minimal implementation requirements and without having to resort to architectural changes to achieve this goal. Indeed, the distributed system framework was thus extended minimally to realise this new adaptive strategy, although the new introductions dictated by Q-learning were substantial.

4.5.4 Q-learning on- and off-policies

The first step was to develop the two different policies, implementing the interface `IPolicyService` for the on-policy (ε-greedy) and the off-policy (Q-function). The on-policy was implemented as a stateful service with the only property being the variable titled `epsilon` so that the parameter can be tweaked as needed for exploration vs. exploitation balancing. This is the first call the new and updated `IIntelligenceService` providing the Q-learning implementation makes, a straight-forward implementation. The on-policy creates a random number and compares it to the `epsilon` value to decide whether one random action from the existing *Action Space* will be called, or the off-policy will be called to provide the optimal action instead. This uses the space created for the *combined* strategy: listen, solve the front-back problem, move in one of 8 compass directions one tile or two – random pick among a total of 18 possible actions.

The off-policy, the one being improved, was constructed as another stateful service with a simple variable as detailed earlier – the `QTable`. A simple interface method for updating the policy was added, implementing the reward function as per the Bellman Equation adapted to Q-learning, which will be presented next. Regarding the table variable, it was made to accept actions as columns that are a fixed 18 in total representing the actions that can be taken, while the rows that would be inserted as encountered represented states that are of an unknown total number but discrete in manner, not continuous *State Space*. Encountering states happens from how the agent starts and explores, specifically the starting point being a state that provides another 8 encountered states (i.e. the adjacent tiles in the 2D environment), initialised with a 0 as Q-value and updated after the reward function runs if the specific cell is selected as an action through the off-policy.

These changes alter the microservice interaction in the reasoning layer of the agent. The new IIntelligenceService interaction flow for this new *Adaptive* strategy is depicted in Figure 4.8.

4.5.5 Q-learning action and state spaces

The *Action Space* was the same as for the previous implementation, since the agent possible actions do not change (i.e. no new sensors or motor functions introduced). The *State Space*, on the other hand, required further investigation due to the problem of having to limit the number of states in the space for a *Discrete State Space* algorithm such as Q-learning [184]. At that moment, the states were not designed with RL in mind and how many states needed to be introduced to make a feasible space, because it has a high impact on the algorithm, and their design could even be used for a *Continuous State Space* policy instead.

The new states, titled `QState`, can be uniquely identified by:

`ObstacleNorth` Values `true` or `false`.

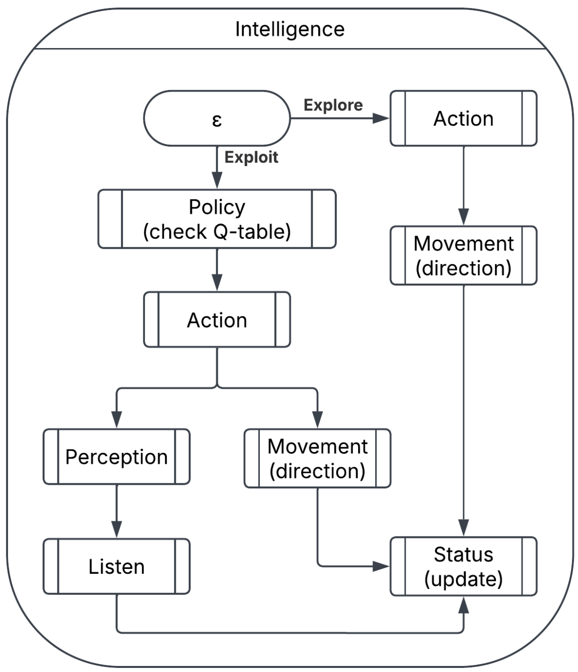

Figure 4.8: The new interactions between the microservices as initiated by the decision-making layer of the agent in the new *Adaptive* strategy.

`ObstacleEast` Values `true` or `false`.

`ObstacleSouth` Values `true` or `false`.

`ObstacleWest` Values `true` or `false`.

`OrientationTracker` Values `N`, `NE`, `E`, `SE`, `S`, `SW`, `W`, `NW`.

`OrientationTarget` Values `N`, `NE`, `E`, `SE`, `S`, `SW`, `W`, `NW`, `U` (unknown).

The above list can identity where the tracker is facing, where the target they had localised appears to be at, and what obstacles are around them. This could assist the agent in devising their strategies for obstacle avoidance depending on their current position in relation to the perceived position of the target [237], which is crucial to minimising energy loss and time spent tracking. This is a representation of the environment free from artificial or observer knowledge that only considers what the perception mechanisms of the agent are capable of, thereby ensuring that the training takes place under the conditions and restrictions that would be imposed to the agent in a real-world application, too. This design evidently limits the *State Space* to $\mid S_i \mid = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 8 \cdot 9 = 1152$ total states, and by consequence rows in the `QTable` with a resulting, cell count (i.e Q-value count) of $\mid Q_{(S_i, A_i)} \mid = \mid A_i \mid \cdot \mid S_i \mid = 18 \cdot 1152 = 20,736$.

The reason behind choosing this approach was to teach the agent how to minimise costs by best avoiding obstacles on their way to their localised target. Introducing more variables to the design of a single state or keeping some of those developed for the *combined* strategy, would highly increase the number of state-pairs an agent can explore to numbers that can make simply training simulations prohibitive. This is where a Deep RL approach could be used instead to get better results [177], or at the very least function approximation techniques can be designed and realised to estimate the Q-value of the unknown state-action pair [238], both of which could eventually result in a more efficient strategy.

4.5.6 Q-learning rewards and penalties

The final piece of the puzzle for development purposes was to implement a reward mechanism. This is a key step in every TD algorithm such as Q-learning, as it happens at every single time interval where the agent transitions into a new state instead of just receiving delayed rewards at the very end. In the case of energy (i.e. battery) efficiency in optimal control theory and similar fields the cost function is generally described using penalties instead of rewards to ensure that the agent is taught that every step of the way incurs a penalty. For Q-learning that capitalises on positive rewards, similar cost efficiency approaches introduce the addition of a bias value to ensure positive rewards. The reward function in such cases is analogous to the cost function in conventional optimal control theory, thereby offering a well-established link to the energy optimisation effort [239].

Designing the function, the energy lost for the action taken is the main factor here, compared to the total energy still available to the agent. The total energy needs to be included in the function so that its weight is reflected in the training of the agent, given that it was not included in the *State Space* design. Unfortunately, that could create a continuous instead of a discrete space introducing huge problems to the chosen RL algorithm and possibly require reliance on more advanced methods that would expand beyond the scope of this project. Moreover, there is a need to factor the time-to-target in the reward function, as it is the secondary factor in the energy-efficient metrics for the results. The bias introduced to elicit positive rewards, as expected of the Q-learning algorithm training process, is a multiple factor of the average action cost in the action space of the system and the time required to take an action (i.e. the *stop-perceive-act* interval) – this is bound to change depending on different physical-to-logical mapping of energy values and motor spin times.

In consequence, the reward function uses a bias with the energy lost per interval over the total energy to enable energy maximisation with positive rewards:

$$R = b - \frac{E_A \cdot \Delta t}{E_{max}}$$

R	The reward for the action taken.
b	The constant bias to guarantee positive rewards.
E_A	The energy cost for the chosen action.
Δt	The duration in time for this action.
E_{max}	The total energy available to the agents.

The time duration is fixed as designed for $2s$, which represents the stop-perceive-act cycle of the agent. Same for all actions except for moving 2 tiles when applicable, which is double that. The maximum energy available to agents and energy costs have been described when discussing the concept of energy for this study earlier, as the product of the very earlier experiments with bio-inspired strategies and evaluating the system. Bias is used as proposed by similar studies by [239].

4.5.7 Policy training - hyperparameter tuning

As a first step in the policy training phase of RL, this section revolves around the values for the triplet of $\boldsymbol{\varepsilon}$-$\boldsymbol{\alpha}$-$\boldsymbol{\gamma}$ that have impact on both training and runtime for the parameters. As mentioned earlier, these parameters are interconnected to various degrees and changes in one also affect the expect outcome of another, or its expected impact instead. Therefore, each individual hyperparameter will be revisited and discussed in the context of how it was utilised to train the *adaptive* strategy before it can be put to the test in a real environment.

Learning rate ($\boldsymbol{\alpha}$) The learning rate governs to what extend the values in the off-policy are updated for each action taken during a training session [240]. The learning rate is usually designed to be a non-stationary parameter, much like the Q-values are also non-stationary. Consequently, an eventual degradation from an initial value across training sessions for this value is desirable [186]. As outlined before, higher rates mean quicker learning but also non-optimal policy formulation, whereas lower rates extend the development of such a policy but nearly guarantee that it can be reached. This guarantee has been mathematically proven when each state has been revisited an infinite number of times, although in practicality this means that at least each state-action pair has a non-zero value for extremely large state spaces [184].

Deterministic environments can benefit from high learning rates, however dynamic environments should gradually start from high and move to low over several iterations [186]. While there can be a delay in reaching the optimal policy towards the end when the rate is significantly decreased, this is a process that is bound to ensure that a more optimal policy for adapting to the environment under any circumstances can be formulated. In fact, the type of degradation itself can result in either polynomial or exponential time costs in reaching convergence as a factor of the discount rate $\boldsymbol{\gamma}$, with polynomial vs. linear degradation rates [186]. With the former being more suitable for training, the concession is that the learning rate should start high and decrease at high intervals.

Exploration rate ($\boldsymbol{\varepsilon}$) Exploration rate has an analogous effect in training to the learning rate [185]: higher rate allows more exploration and population with less than optimal values in the Q-table, with the lower rate exploiting more and thus learning to choose better actions over time. The difference here is that exploration rate is also something that factors in the final performance of the algorithm, too, apart from the training, while having different roles in each: in the former it helps with escaping local optima in decision-making, whereas in the later it allows for fine-tuning existing Q-values or filling in missing ones [186, 185].

In consequence, analogous to that of the learning rate is also a gradual degradation of a starting high exploration rate for training. In the beginning it is required to employ high e to populate zero-value Q-table entries, with a view to improving them in the future with a lower rate. What a high $\boldsymbol{\varepsilon}$ also offer to the table in training is the reduced risk of missing optimal action by relying on sub-optimal known exploitation options instead [186, 185]. Accordingly, while the training of the agent in the *adaptive* strategy evolved, so did the exploration rate decrease in a manner parallel to the decrease in learning rate.

Discount rate ($\boldsymbol{\gamma}$) Discount rate differs from the above in its actual effects and plays other important roles in the fine-tuning of training hyperparameters. Lower rates ensure the agent values short-term rewards during training, nonetheless the focus of this strategy is to create one that can adapt in the various environments by reaping high rewards in the end. Given the R function described earlier, this would entail the lowest combination of cost and time, thereby ensuring the best possible time and energy saving with a view to the eventual goal of locating the target. Evidently, the training would require setting a higher $\boldsymbol{\gamma}$ value to ensure that the whole training process is centred around always maximising the best rewards in the long run.

Unlike the learning and exploration rates, there is no gain for training with varied values across the whole process [186]: a lower rate would only establish not exploring "hidden paths" to better rewards serving only to explore in local minima and improve the values there for short-term goals. Another advantage of a high discount rate value relates to the possibility of enforcing a form of exploration without relying on the exploration rate – to elaborate, this occurs due to learning to skip the discount rewards to find these more optimal "hidden paths" to the long-term goal [186, 185]. Moreover, similar research in battery cost optimisation utilising Q-learning and focusing on reward maximisation in tandem with the research on learning rate costs and optimal hyperparameters [239, 241] , reveal that discount rate factors for such cases should waver between 0.7 and 9.0, with the latter being the advised choice [186, 241, 239]. Accordingly, this value will be set for all experiments.

4.5.8 Policy training - episodic training

What is described as an *episode* is the deployment of an agent through the baseline scenario with specific hyperparameters to populate the Q-table with Q-values were missing (i.e. having a value of 0), or to improve those values with new ones owing to a new set of training hyperparameter combinations. Given that the agent will need to be able to adapt to a highly dynamic environment, this is the baseline scenario that the agent was trained in: few obstacles raising to high during operation, and one initial source with the additional "noise" of more sources being introduced gradually. Through this scenario the agent could learn to tackle obstacles, find routes even through new obstacles, and reach the desired target as needed.

Therefore, every episode consisted of this scenario and the agent running through it until completion (i.e. depleted energy or target reached), updating the Q-table along the way with each *episode step* (i.e. taking an action). Consequently, the table was serialised in a condensed binary format given its size and was reloaded at the beginning of the next episode so that the values could be updated. Lastly, the different *episode sets* refer to the assorted combinations of hyperparameter values that ran through several episodes each. For each such *set* some information had to be gathered before moving on to the next set, to keep track of possible algorithm convergence through proper metrics.

One of the most prominent tools for determining algorithm convergence in machine-learning is the use of Root-Mean Square Error (RMSE), a tool primarily used as a regression model performance indicator:

$$RMSE = \sqrt{\sum \frac{(Pi - Oi)^2}{N}}$$

Pi The predicted choice at each step (i.e the optimal path if it were known).

Oi The observed choice at each step (i.e. the choice the agent made then).

N Total number of episode steps taken.

To enable the system to be capable of providing a value for Pi there needs to be a way to ensure what the best possible choice would be at each step. In this case, the focus is on finding the best next step rather than cost, where a Dijkstra uniform-cost search would be more appropriate. This can only be provided in theory by the observer and to achieve this goal the use of Breadth-First Search (BFS) was necessary, an algorithm specialised in finding the minimal path from start-to-end in a grid-like environment with obstacles, as compared to Depth-First Search (DFS) for example [242]. The implemented algorithm for BFS provides the best possible path to move into towards the target, which is a set of state-action pairs where the action Ai there is a single move action, still that is the only capability it offers.

BFS provides the path as a list of Q-table cells, from which eventually more processing is required: first to determine if listening is needed, which the `IEnvironmentService` determines if the target moved in the last step and is one compass direction farther than the agent, and second if two consecutive steps can be taken towards one direction instead of just one to reduce time. Lastly, these comprise the next best prediction and thus the Pi, while the Oi is the state-action pair the agent eventually chooses. Apparently, this is a very costly step in computational power to perform at each episode step, which cumulatively builds up to a much higher training runtime for an episode and by extent a training set. As such, this metric has only been used to observe convergence capabilities of the *adaptive* strategy only during the last two sets of hyperparameter combinations that revolve around exploration rate $\boldsymbol{\varepsilon}$ and will be of consequence to the eventual experiment results that will be showcased next. The use case is the highly-dynamic scenario with only one tracker, the one being trained (i.e. with the adaptive strategy).

The following *episode sets* emerged and followed in order where $\gamma = 0.9$ for each:

ES1 $\{\varepsilon = 0.9, \alpha = 0.9\}$

ES2 $\{\varepsilon = 0.6, \alpha = 0.9\}$

ES3 $\{\varepsilon = 0.3, \alpha = 0.9\}$

ES4 $\{\varepsilon = 0.6, \alpha = 0.6\}$

ES5 $\{\varepsilon = 0.3, \alpha = 0.6\}$

ES6 $\{\varepsilon = 0.3, \alpha = 0.3\}$

ES7 $\{\varepsilon = 0.1, \alpha = 0.3\}$

ES8 $\{\varepsilon = 0.3, \alpha = 0.1\}$

ES9 $\{\varepsilon = 0.1, \alpha = 0.1\}$

In closing, the very last set **ES9** was, naturally, the one to be used on the experiments as the aspiring optimal and final *adaptive* strategy for the agents, nevertheless **ES8** has been persisted as well, for investigation through the developed BFS-RMSE convergence metric for evaluation purposes. Numbering $5,000$ episodes each, due to the average number of *episode steps* for each episode being enough to cover the entirety of the *state space* $\mid S \mid$, thanks to the exploration randomness at least for the start.

4.5.9 Policy training - training observations

This section presents some noteworthy observations related to the **ES*** iterations and how the agent started to learn how to solve the tracking problem step by step. To begin with **ES1**, the most important part of this set was to explore vastly and aspire to populate all Q-value states with a non-zero value as per the R function. Indeed, the results showcased a coverage of about 100% of the Q-table cells, which was also continued with more optimised values through **ES2**. These initial training sets provided the based for the rest of the training process to iterate and evolve into the coveted optimal policy.

Moving on with training sets **ES3**-**ES4**, the most interesting observation as Q-values were improved from episode-to-episode from these initial sets was that the agent appeared to learn, that the first thing they should do at each initial state is to attempt and localise the target. The initial state of the episode always sets the `OrientationTarget` value of the state S_1 to `U`, which is the unknown location of target. Evidently, most attempts where the first exploration step of such resulted in anything other than listening for localisation led to eventual energy starvation due to not knowing where to head towards. This proves in practice beyond just theory for the CASA tracking case at hand that Q-learning can provide a valuable strategy towards solving the energy-efficiency problem.

Consequent attempts **ES5**-**ES7** proved once more the Q-learning fact: the agents started to learn to follow walls when a target was on the other side of that wall. Instances between attempts were investigated by parsing the Q-values for specific similar action-state pairs. The pattern that was identified had to do with similar `Obstacle*` values aligning on the $Q\left(Si, Ai\right)$value where `OrientationTarget` value was within one step of that: an example of `ObstacleNorth` is true and `OrientationTarget` is `NW`, `N`, or `NE` for consecutive Si states, examples of which can be found in Figures 4.9 to 4.11. This can be argued to be the outcome of a high $\boldsymbol{\gamma}$ value teaching the agent to follow such consecutive states with the eventual high reward of finding the target at the end. This something unique in all the strategies implemented up until that moment, as the original strategies did not have any elaborate obstacle tackling mechanism, merely a reminder of where an obstacle was just to not go

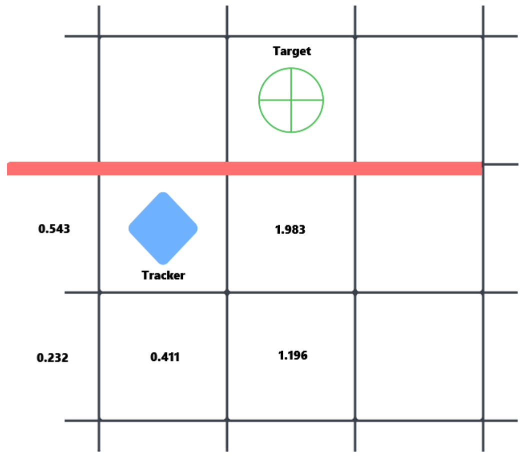

Figure 4.9: Rewards for possible movement actions for the optimal policy (highest reward will be chosen). Higher values for going closer to the target and closer to the obstacle.

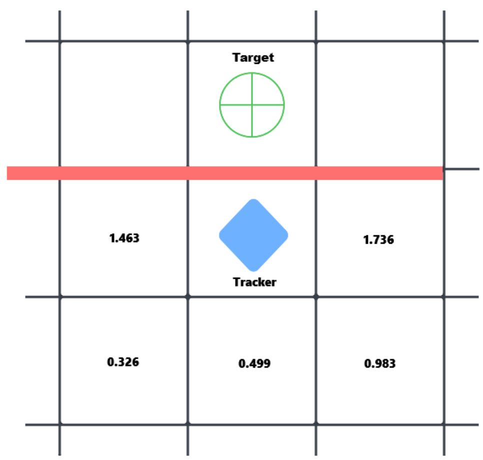

Figure 4.10: Rewards for possible movement actions for the optimal policy (highest reward will be chosen). Higher values towards grids away from past obstacles and closer to target. Notice how previous step is still high, as it means following an obstacle, but due to the current orientation of the head (i.e. East) the value there is much higher.

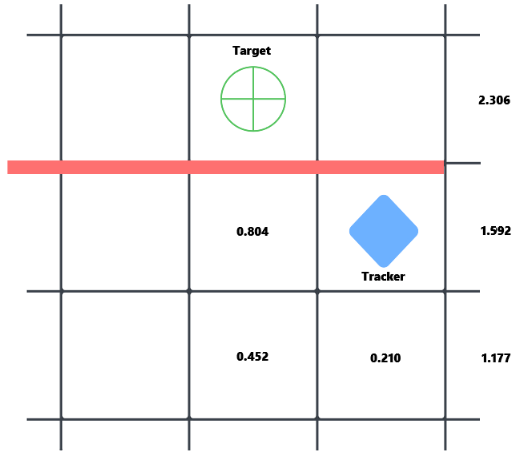

Figure 4.11: Rewards for possible movement actions for the optimal policy (highest reward will be chosen). Higher values demonstrating how the agent eventually learned to overcome obstacles.

near it in short-term exploration. Eventually, this comprises one of the gains of machine-learning over the more basic bio-inspired strategies.

At the same time, a similar observation was made with regards to the environment open-space sprint towards a target, which is characteristic of the *explosive* strategy. In this scenario we have the pure AI algorithm learning to behave as a treefrog under favourable explosive breeding conditions merely by being given the choice of having double-move as a potential action in the action space A. Indeed, states where both target and tracker where in the open space and the target orientation did not deviate more than one step on the compass showcased high $Q\left(Si, Ai\right)$ values. One more key takeaway from this observation is that the agents learned to also value the time to reach the target apart from the energy cost. In detail, the double-move action has much higher cost, but the potential of reaching a target close by in open space just by following two consecutive double-move actions greatly increased the final rewards as compared to the immediate ones, another benefit of training with a high $\boldsymbol{\gamma}$ value.

The final two sets **ES8-ES9** did not have anything ground-breaking to showcase as an observation, but fundamental information was observed through analysis of the BFS-powered RMSE metric for convergence. The combination of values $\{\alpha = 0.1, \gamma = 0.9\}$ in these sets naturally assure the proper pace of learning and the significance of the future reward. Indeed, some training processes, nonetheless in less dynamic environments with smaller state space, prefer to just start with these values and run a significantly higher number of episodes until convergence can be observed.

Table 4.4 demonstrates the values of RMSE for the last two training sets that

	RMSE (25%)	**RMSE** (50%)	**RMSE** (75%)	**RMSE** (100%)
ES8	0.8597	0.8543	0.8441	0.8398
ES9	0.7991	0.7911	0.7889	0.7877

Table 4.4: RMSE values over percent of total training episodes (5,000) completed.

were of importance. Naturally, RMSE values indicate better performance the closer they are to 0. Relevant stability across and between the last two episodes also indicates that there were no significant gains in training sessions, therefore more stable values were being achieved. Accordingly, these values indicate that potentially more episodes might be required to provide a full convergent policy, however these values are close to what an optimal policy fluctuates within and thus could be utilised towards evaluation against other strategies with **ES9** provides the best performance.

4.5.10 Experiment designs

Having reached the point of an *adaptive* strategy trained to sufficient length, the time was ripe for answering the research questions. The experiment designs have been detailed in the respective section that corresponds to the *combined* strategy and, given the similar questions for that set of experiments, there is nothing to change or add here with new designs are needed towards answering the research questions pertinent to this iteration – including **RQ8**. The only two factors that changed at this point were the strategy that the agent employs at any time, that is the fully-trained *adaptive* strategy, and the exploration rate.

The final experiment list was shaped as:

AF* Adaptive strategy, few obstacles, * targets.

AM* Adaptive strategy, many obstacles, * targets.

BF2 Both Adaptive trackers, few obstacles, two targets (locate both).

BM2 Both Adaptive trackers, many obstacles, two targets (locate both).

HDA Highly dynamic environment, one adaptive tracker.

For the purposes of the experiments a trained policy is used, which implies that the triplet of $\boldsymbol{\varepsilon}$-$\boldsymbol{\alpha}$-$\boldsymbol{\gamma}$ that governs the eventual behaviour of the algorithm is now obsolete. Indeed, it is now condensed down to the single value of $\boldsymbol{\varepsilon}$, which factors into the normal operation of the Q-learning algorithm. The exploration value in highly and meticulously trained policies, which have been deemed to have achieved convergence and generally operate in non-dynamic or low-dynamic environments, is usually set to values below 0.01. This is to ensure that the optimal decisions are made at each point during the runtime operation of the agent towards solving a real problem and not during a training episode.

	Success Rate (%)		Energy Left (%)		Time-to-target (s)	
ε	0.1	0.3	0.1	0.3	0.1	0.3
AF1	93	91	35	22	59	65
AF2	94	90	36	21	55	66
AF3	95	89	39	23	51	69
AM1	91	86	29	18	65	72
AM2	92	88	31	19	63	74
AM3	92	87	30	19	60	70
BF2	91	83	19	14	84	94
BM2	90	85	17	12	88	98
HDA	88	78	18	8	72	85

Table 4.5: Success rates, energy remaining, and time-to-target rounded to closest integer for the *adaptive* strategy experiment runs, for both values of **ε**.

Taking into consideration the dynamics of the environment and the less extensive training of the policy, which is also not supporting with more advanced techniques such as Deep RL, a value much higher is needed to ensure that the agent will not get stuck in local optima or obstacle loops. In such cases, a value of **ε** of at least 0.1 is used as discussed in training, prompting the experimentation with different values to answer the research questions. To this end, the values of 0.1 and 0.3 were chosen for 2 runs of the experiment sets using the fully trained policy as resulted from **ES9** but with a differing **ε** value. The chosen value of $\varepsilon = 0.3$ is also close to the value of movement in the *explosive* strategy given that exploration means movement one out of five times but using the fully trained policy this time.

4.5.11 Results

Results for the final experiment collection towards evaluating the *adaptive* strategy compared to the bio-inspired and *combined* strategies were critical in determining its viability for the purposes of this thesis. Many results to compare to were available, which only served to bring to the spotlight the performance gains of this strategy: vastly outperforms the bio-inspired strategies, while at the same time outperforming the *combined* one by a significant margin. The important results are organised in Table 4.5 (a high-level overview). A presentation of results in the context of success rate, energy and time efficiency will be performed once more gathered through the (T, E, S, O, M) logging. The *combined* strategy had offered the best results up to this point to compare with (Table 4.2). A quick look at Table 4.5 side by side with Figures 4.12 and 4.13 reveals that $\varepsilon = 0.3$ provided absolutely no benefits in any cases as compared to $\varepsilon = 0.1$, therefore the results will be presented for the latter only.

Regarding **success rates**, it is clear that the values are staggeringly higher in all base experiments with few obstacles (ranging from $+16-24\%$), and even

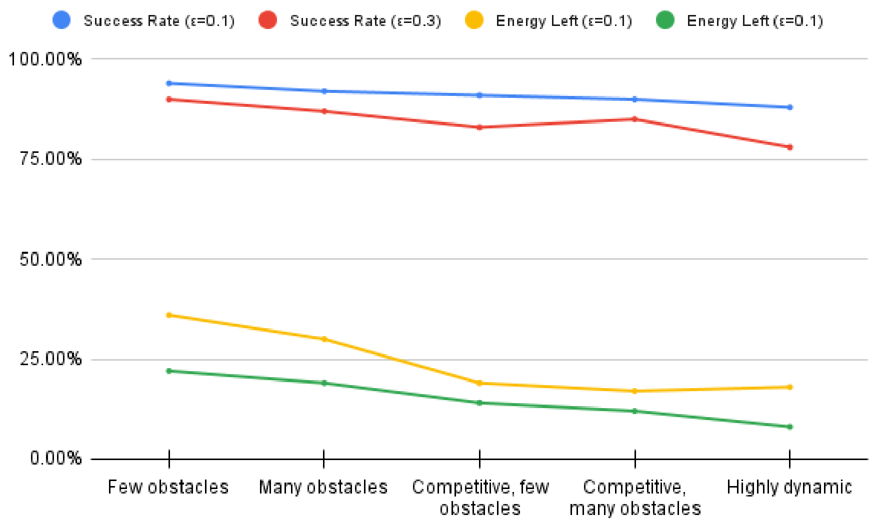

Figure 4.12: A graph comparing the metric percentages for the *adaptive* strategy in the different environments it was tested in for both values of $\boldsymbol{\varepsilon}$.

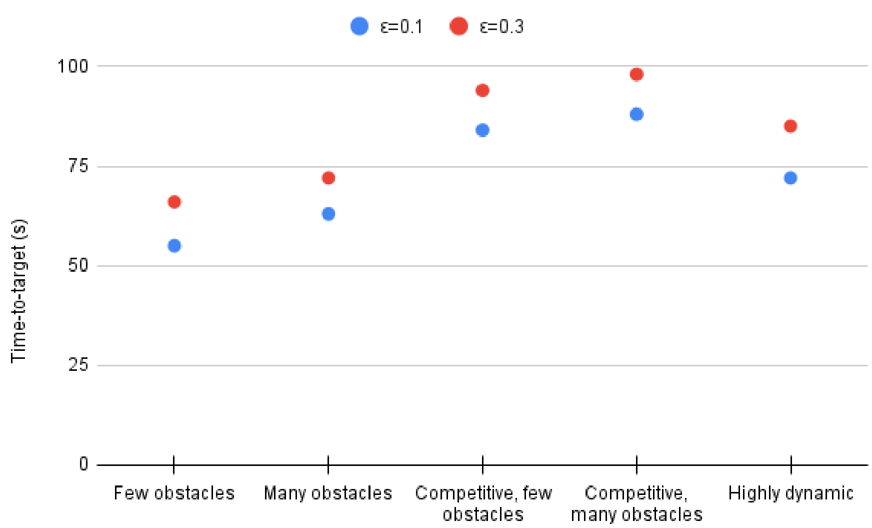

Figure 4.13: A scatter graph representing the time-to-target for the *adaptive* strategy in the different environments it was tested in for both values of $\boldsymbol{\varepsilon}$.

higher compared to the boosts gained from the *combined* strategy when a larger amount of obstacles is involved in the scenario (average gains of +28.3%). Another noteworthy observation with regards to success rate for the competitive tasks (**BM2**) is that once more the gains were far higher than the combination of *explosive* and *regular* strategies, or the *combined* alone, with a gain close to +9% to either of the two. Final substantial gains come with the highly dynamic environment experiments (**HDA**), where the gain amounts to an impressive +22% over the *combined* strategy, with the *bio-inspired* ones not offering any competition as discussed in the previous chapter.

Meanwhile, **energy efficiency** appears to also be improved in most cases and, especially so in the experiments with few obstacles this time instead, but more importantly the gains seem to be somewhat analogous to the gains displayed for the *success rates*: +17% for more as opposed to +19.6% for less obstacles, surpassing the highest numbers so far that the *regular* strategy has exhibited by almost as much. The increase holds true for the base test cases but not the competitive tasks, admittedly, where the gains were just lower that a flat 10%, producing naturally better results than the combined strategy but not nearly as impressive. In **HDA** (a +11% increase over *combined*), notably, the energy efficiency gains for $\varepsilon = 0.1$ are almost double in contrast to the $\varepsilon = 0.3$. In fact, the latter has had much lower gains in energy efficiency than in success rate unlike the former.

The most distinguished observations come with the **time efficiency** results, due to the uniformity they represent across scenario families (62.6s for many obstacles, 55s for few obstacles). Moreover, it is evident that *adaptive* is the very first strategy that produced consistently high success rates at moderate time costs, thus overtaking anything the *regular* strategy had to offer to the problem-solving with also similar energy costs and much higher success. Time efficiency is very close to that of the *explosive* strategy, nevertheless not only for the more open spaces where that strategy shines. The most important gains here are also on the **HDA** experiments, where investigation of time-to-target showcases roughly $^2/_3$ of that of the *combined* strategy. Lastly, similar gains were found in the competing experiments, but merely of a factor of $-10s$ average.

4.5.12 Discussion

The first question that can effortlessly be answered through the results above, and their proximity to optimal values, is **RQ8**: indeed, an *adaptive* strategy based on Q-learning and trained extensively with degrading hyperparameters can provide a solution to the energy-efficient mobile audio source tracking problem. With the high success rates and a balance of energy to time ratio towards finding a target under any environment, whether less or more dynamic, an agent trained in this manner can adapt to the problem and find an efficient solution in a timely manner. Consequently, most of the research questions posed can be provided the exact answer (**RQ1-RQ4**), except perhaps the competing set of experiments where no significant gains could be observed (**RQ5-RQ6**). Lastly, the high adaptability demonstrated for the highly dynamic environments is also the much sought-after answer to **RQ7**.

Ultimately, the more concise discussion for these findings in this section focuses on the observations and how they are tied to the RL approach. One interesting finding was pertinent to the performance in dense environments. This can be assigned to the policy training observation highlighted earlier: how the agents learned to follow long walls in overcoming obstacles. Its significance is attributed to the recursive division algorithm employed towards simulating a floor- or apartment-like environment does indeed tend to create at least one larger corridor, but also both spacious and smaller rooms, simulating a much more space. Naturally, the trained algorithm could adapt in such a manner to actual real-world spaces that bare such characteristics. Nonetheless, the importance of being able to efficiently tackle obstacles in a dense environment is essential in disaster scenarios, further increasing the applicability of the *adaptive* strategy in demanding cases.

Another fact that emerged through observation of the results and a brief analysis of choices made by the agents after running isolated experiments, is that the agents learned to balance listening and moving to get the best results in both energy and time costs, thus reminiscent of *regular* strategy traits. The investigation showed that the behaviour with which the agents were trained learned to stop and listen after somewhere between two and four steps taken depending on the compass direction distance in their orientation from the orientation of the listener. This ensured they did not move more than needed towards the direction of a target before that target had changed their position or they have moved much further from that position. It is speculated that the long term-reward emphasis through the high discount rate suggested for such cases contributed to this behaviour.

Investigating the results for the most open-space (i.e. low-obstacle environments) tracking showcased the adaptation of the capabilities of the *explosive* strategies towards finding the targets, albeit in a more sophisticated manner. The previous observation also could be parallelised to the more intricate stop-and-listen behaviour of *regular* strategies, still once more in a more expert manner accounting for other important factors, too. This highlights the fact that the idea of reaching a point of attempting to create a machine-learning AI through bio-inspired approaches contributed to the success of the eventual policy. Indeed, the double-move action would not normally be a choice presented in a design starting from scratch given the actions robots could take by default – simple move is expected to be the only choice by default.

On the other hand, a confounding observation can be made after repeated experiments with the competing trackers (even when possessing the same strategy): any optimisations through better strategies provide minimal gains as compared to other scenarios, even the more dynamic one. It can be argued that the nature of the problem itself, which can be branded as a case of CPS to some extent provided it is required to find all targets, is what contributed to the overall high energy and time costs for all strategies in the end. Naturally, this provides an opportunity to find other ways to solve the problem, obviously through collaboration instead of competition on a personal level, which can be facilitated

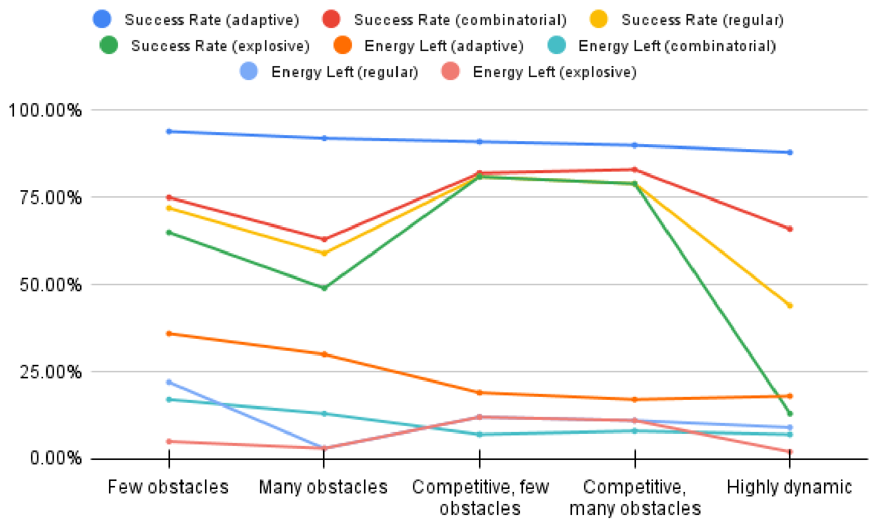

Figure 4.14: A graph representing the metric percentages for *all* strategies in the different environments they were tested in.

through social emergence that is planned to be studied next.

Finally, setting the exploration rate to a value much higher (0.3) than the expected (0.1) for similar attempts did not offer any significant advantages in any of the scenarios, making the default choice adequate as an option. Admittedly, results with a much lower rate (i.e. the suggested 0.01 at most) could have been investigated, too. Nevertheless, the combination of the lack of gains using a much higher rate with the suggestions from background theory that less-than-optimal outcomes can be gained with such low rates in highly dynamic settings, notwithstanding the time constraints imposed by the scope of the study, resulted in skipping this attempt. Having developed, fine-tuned, and evaluated an *adaptive* strategy, the study **Objective A2-O3** has been fulfilled.

4.6 Conclusions

The purpose of the work covered in this chapter sought to focus on developing a strategy for tracking sound sources moving erratically within assorted complex acoustic scenes in an energy-efficient manner. The process was carried out entirely in the developed CASA distributed system framework, despite the lack of use of much of the distributed aspect of it. Nonetheless, it was capable of accommodating anything from simulations to bio-inspired strategies, to artificial strategies, and even machine-learning training in the form of RL. Consequently, this chapter also served as further validation of the capabilities of the distributed framework for CASA applications and research.

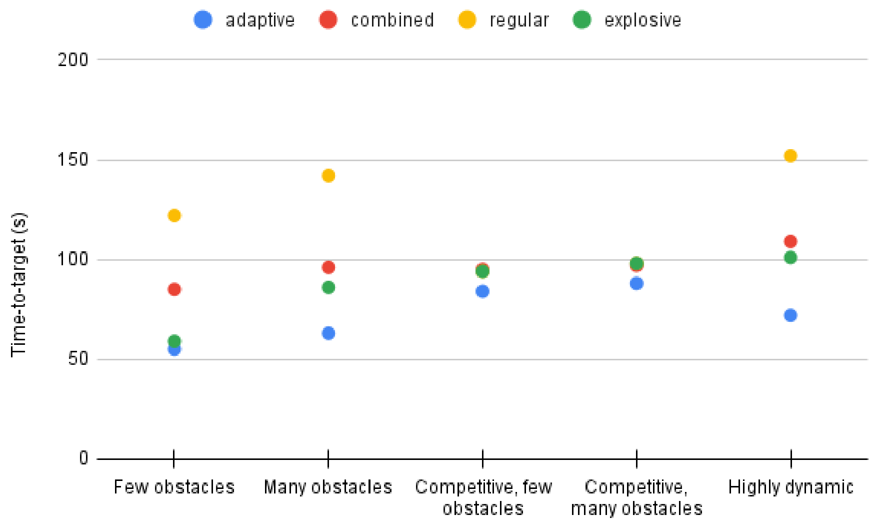

Figure 4.15: A scatter graph representing the time-to-target for *all* strategies in the different environments they were tested in.

The strategy development process was heavily influenced by bio-inspired computing. Indeed, the evolution process for the strategy followed the very distinct steps: modelling the bio-inspired strategies, establishing a baseline from them, leverage any important features and behaviours to develop a more performant artificial approach, and ultimately employ all that knowledge to develop an AI capable of addressing the problem using established machine-learning algorithms and disciplined development processes. Strategy evolution performance with regards to the metrics is illustrated in Figures 4.14 and 4.15.

An argument could be made against following this approach but instead starting right away with designing and training an AI to solve the specific problem. Such an approach would allow for a much more elaborate and involved training process in a more refined stepwise manner, where the strategy is training in even more steps where the exploration and learning rates gradually decrease, or overall including more involved hyperparameter tuning, whilst ensuring the state-action pairs are covered almost in their entirety several times across all iterations. While this could ensure much better results towards ensuring the off-policy in the Q-learning AI could have achieved convergence, there would be missed opportunities that were discussed above towards the whole process.

The first and most important would be the idea to utilise a double-move action inspired by the explosive breeding treefrogs and other arboreal locomotion animals employ, which did provide better performance under specific conditions and, by consequence, in highly dynamic environments when the agents find themselves operating under similar conditions at times. The second would be the baseline that was established by the original bio-inspired strategies, as well as the combined one – these offer benchmarks to compare future work to, given the lack of results from similar research work to evaluate against (i.e. energy-efficient target tracking by sound alone). Finally, the emergent phenomenon of *complementarity* would not have been observed, spurning the idea about combining strategies for a better solution to the problem, especially given a future CPS solution.

Ultimately, the chosen approach resulted in a strategy that can provide an adequate solution to the energy-efficiency for trackers using sound alone in highly dynamic acoustic scenes, thereby allowing this thesis to reach its second high-level **Goal A2**. The twofold contribution that stems from this aim is comprised of the strategy itself, which could be now used with the necessary adaptation to real-world scenarios or further studies, as well as the methodology of how to borrow inspiration from nature or biology and evolve this into highly performant software that can solve complex problems. Furthermore, the whole process towards realising goal **A2** also validated the high-level goal **A1** once more, bringing together the two of the three parts of this thesis and setting the scene for attempting to realise the next aim.

Some limitations of the product so far need to be acknowledged. Particularly, the energy reserves (defining E_{max}) have been chosen arbitrarily, in contrast to the energy costs for actions which have been devised according to relevant metrics. Nonetheless, while the value is not analogous to that of potential real-world hardware applicable to the problem being solved, it has been selected diligently after test runs of the initial strategies. This was to ensure that the values would be enough to support both formulating a proper baseline to compare with for future results, as well as for future problem-solving endeavours to have room for improvement and thus provide substantial results, if possible. The other important issue remains the extent to which convergence has been achieved on the optimal policy of the *adaptive* strategy – whilst the BFS-RMSE evaluation results did reveal the strategy being close to convergence, it has not been investigated to painstaking length. Evidently, the results through the experiments were enough to justify its efficiency and efficacy over the other strategies and in general as a tool towards solving the energy-efficient tracking of mobile audio sources.

These limitations could, eventually, be addressed through future work. Areas that could be touched include but are not limited to: the deployment of the system on robotic devices to measure energy reserves and costs relationship in actual hardware, more extensive training with increased episode count in sets to cover more state-action pairs more efficiently, expanded hyperparameter fine-tuning during training, and rigorous investigation of optimal policy convergence. Such steps could ensure that the strategy has achieved its peak performance and can indeed be utilised towards real-world testing and evaluations. To conclude, the existing results are encouraging and supply a firm foundation to facilitate further experimentation on improving such policies through other avenues, such as through emergence.

Chapter 5

Leveraging Social Emergence

5.1 Introduction

Collaborative Problem Solving (CPS) is a field that involves communication and interaction between entities during learning, problem-solving, or coordinated tasks. Such tasks can be complex problems in complex environments that require the impeccable effort of numerous individuals to solve and naturally result in complex systems to manage. When intelligent agents take over the role of individuals in CPS, the concept takes on a new dimension and results in complex distributed systems of cooperating AI of possibly assorted capabilities, all working together to achieve a goal. Such systems, heterogeneous and interacting, are distributed systems that can foster emergent developments that can either contribute to or detract from the capacity to solve the problem in question.

This chapter is focused on adapting the socio-cognitive traits that can enable *emergence* and its beneficial properties in the system, within the intelligent agents that govern the new behaviour for CPS purposes. The architecture changes in the system and the AI, the research questions, and the experiments related to these preliminary studies with emergence are presented to showcase how the previous results from this thesis can be applied to achieve higher efficiency and efficacy in solving the even more complex scenario of tracking all mobile targets in a highly dynamic acoustic scene. In conclusion, it is imperative to define that the focus on determining emergent behaviour will be reliant on empirical evidence via the experiment results instead of a disciplined and framework that would require not only additional work, but rather even higher degree of complexity to the problem and implementation of tools that can identify such phenomena in a more robust manner.

5.2 The Socio-cognitive Traits

5.2.1 Introduction

Having discovered the traits that can be utilised for improving the CPS capabilities in the system, the designs can be developed to embed it to the existing intelligent agents. Up until this point the only CPS capabilities that have been identified in the system came from the experiments in the *competitive* category where two trackers tried to find all targets in the acoustic scene without any interaction between them. This is where the complementarity emergent phenomenon was also observed that inspired further work in the CPS field. Towards enhancing this potential for solving the more complex issue, the designs of the *adaptive* and *combined* strategies had to be revisited and tweaked to support the new decision-making behaviours demanded by the new traits.

Meanwhile, the EDS framework developed allowed for an interface for communication though no such implementation was made, which is something else that will be showcased, too. This is part of the *social* traits being introduced along with the need for reliance on external input from other agents to determine what they should be doing or if they are on the right track. Cognition plays a larger role in the decision-making of the agents, naturally, so the new *cognitive* traits of the agents are detailed here. These include the degree of certainty an agent "feels" they have for their current tracking task, and which affects their reliance on *social* interactions, as well as the strategies that the agent will employ to track their targets.

The distinction implies a new separation of concerns for the agents, the decision-making for what to do generally and the decision-making if they need to track their target. Naturally, the latter is the part where the *adaptive* and *combined* strategies are evoked, whereas the former is governed by the new implementation that accounts for both new *social* and *cognitive* traits, and their *low* and *high* states. These traits are, evidently, the micro-properties introduced to the agents that aspire to provide new and interesting results through emergent phenomena for solving the problem of tracking all mobile sound sources in a highly dynamic acoustic scene in the most energy-efficient manner.

5.2.2 Cognitive traits

The *cognitive* traits introduced to the agents aspire to provide, or rather to further ameliorate, the dimension of critical thinking capacity. The agent needs to be able to make decisions based on their *confidence* for how well they are tracking their target, while at the same time using their own personal chosen strategy to track. The existence of the implementation for the two strategies that will be utilised to this end greatly reduced the effort towards realising this step. There are several new concepts and aspects of this endeavour that need to be defined and discussed here, pertinent to the selection of the strategies and what *high*/*low* cognition means in the context of this thesis, and similarly for communication.

Notably, the traits covered here will be part of the state the agent knows about itself and not in the state of the environment, as is the case with machine-learning with RL studied earlier. Naturally, the core inspiring social study for working with the *high*/*low* binary values is [211], which offers a baseline to compare to alike the treefrog biological studies.

5.2.3 Base cognitive traits design

Cognition variance between the values *high* and *low* in this context refers to two aspects as also derived from the psychological studies: (a) how capable they are of solving the problem on their own, and (b) how confident they can be about their capacity to do so. With the results of the latest round of experiments dealing with the development of the *adaptive* strategy, it is obvious for aspect **a** that the most performant strategy is that one and hence assigned to *high* cognition, whereas the runner-up (i.e. *combined*) is pertinent to *lower* cognitive skills. This mapping is simple to implement provided the most demanding task is already solved. As such, the trait of `ProblemSolving` is introduced with binary choice of values being *high* and *low*, thereby mapping respectively to *adaptive* (optimal) and *combined* (less-performant) strategy.

On the other hand, and in relation to aspect **b**, the situation is more complicated as observed by psychologists. Most subjects that are not confident in their problem-solving skills will seek help from peers, whilst a smaller number of individuals tend to have a false perception about their problem-solving skills, be it due to a higher developed degree of egotism, perceived self-worth, or other behavioural elements. Breaking down the traits into that many smaller to cover all cases would become a detriment towards the conclusion of this thesis within its scope, while at the same time it can be argued that the reason behind the choice would not matter in finding an optimal solution to the CASA tracking problem, rather possibly further associations between behavioural traits and performance in such scenarios which could contribute more towards the field of psychology. Evidently, this ordained the introduction of a single `SelfConfidence` trait in the state of agent itself, with binary choice between values of *high* and *low*.

`ProblemSolving` in tandem with `SelfConfidence` have been implemented in the system as inheriting the `IState` interface (defines an agent state) the agents employ to make decisions and hence creating the new `CognitiveState` class. These two values will be considered during the first step when the agent "wakes up" in its *stop-perceive-act* cycle and when they attempt to decide what to do next. This process will be detailed later, as the values of high and low need not necessarily translate into binary decisions. To be able to define that the social traits need to be discussed next.

5.2.4 Tracking confidence design

The final part of cognitive implementation comes in the form of the *tracking confidence*, which is entirely different from the `SelfConfidence` trait – in fact, it is not a trait whatsoever. This is essentially a function that does consider the

`SelfConfidence` trait of the agent and compares it to its own perceived *recent* tracking performance. Naturally, the agent does not have a perception of the wider environment to be able to ascertain with a high degree of confidence that they are doing it right, which would defeat the purpose given the restrictions imposed upon the tracking tasks. Accordingly, the agent, through its limited view of the world, ought to be able to determine how well the tracking is going.

Without overloading the agent with information, what it already keeps and knows from the state of the environment can be stored and used. This includes the following: the perceived orientation of the target, own perceived orientation, and recent actions taken towards tracking. In elaboration, the agent attempts to ascertain that they are on the proverbial "right track" by checking *how often* they had to change their orientation and by *how much* during their recent *movement* actions. This is possible thanks to the values retained in the `CognitiveState` type of property of the agent. Lastly, this confidence is further modified by the value of `SelfConfidence`, thereby accounting for this important cognitive trait.

This manages to utilise the much-needed local state of agent without reliance on central authority for information, hence keeping functions distributed in nature [243]. The three values – confidence, frequency of and deviation by orientation changes – have equal weight on the outcome. Inspiration for this approach and the manner the function was designed was taken from pertinent research found in [244, 245, 246], and more specifically the tracking confidence displayed to other being also based on self-confidence apart from what could affect it in the realm of mobile audio source tracking.

The function is designed as follows:

$$C = c + \frac{1}{O} + \frac{1}{\frac{\sum A_i}{n}}$$

C	Tracking confidence.
c	Tracker self-confidence (value between $0 - 1$).
O	Times tracker changed orientation recently.
Ai	The perceived target orientation changes so far.
n	Number of perceived target orientation changes.

5.2.5 Social traits

The core of emergence is the interactions among intelligent entities in complex systems, as witnessed when the first bio-inspired strategies competed in experiments only to highlight how they complement each other, still even more so when this interaction is facilitated through meaningful communication. The aim of introducing communication among agents is at the centre of this attempt so that better performance can be gained. Psychological studies for humans have also highlighted how CPS helps the less capable members of a community to perform better, therefore this is something to look forward to in the results. To start realising this behaviour, the `ICommunicationService` must be extended first, followed by the traits that govern the social behaviour of the intelligent agent.

5.2.6 Communication microservice changes

At the time of the implementation of the *social* traits, the only capability the `ICommunicationService` offered relates to the *status* of the agent, as described in the corresponding section earlier. What this does exactly is allow the agent to ask for the status of another agent, or to reply with the current state of the agent from what it knows (from `IStateService`): energy left, target being tracked. The extended `ICommunicationService` needed to be able to convey more information about the world as the agent knows it, thus transferring information about: (1) the details of a specific known target (i.e. frequency, pulse rate), and (2) the tracking confidence of the agent towards tracking its own target.

Regarding **1** it is essential so that communication of details of the current and preferred targets can take place, which can enable an agent to decide whether they can switch their preference to follow the details of that target. Information piece **2** can be beneficial for a tracker receiving this information from another one so that it can decide: if my co-tracker is so confident in tracking their own target, should I keep following that one or switch to another? The combination of these two data types that can be transferred over the EDS serves the purpose of enabling them to make more complex decisions at runtime to contribute in a more effective manner to the CPS attempt. The above data types can be used in RESTful requests for the asynchronous API call communication between the agents. To enable such communication new endpoints were created that agents can utilise to achieve the higher-level social functions of sharing and requesting information.

The data shared for both requests is common and transfers information about the details of the target, and an associated `TrackingConfidence` value, as per the function output from above when the request takes place, for each of these targets. This information for each `Target` includes the estimated frequency, pulse rate, and perceived positional orientation in the environment (8 compass directions). Of course, an empty such value denotes a target that is not actively being tracked and, given the fact that the agents are tracking a single target, it also conveys the current target of the agent. This should be enough to leverage decision-making on the side of the agent, while at the same time allowing for the implementation of the *social* traits. Figure 5.1 captures this design in a UML class diagram.

These endpoints in the Minimal API are designed as:

`GET /targets` Returns the targets the agent is aware of. Only for agents with enabled social traits.

`POST /targets` Receives the targets from another agent that shares it. Only for agents with enabled social traits.

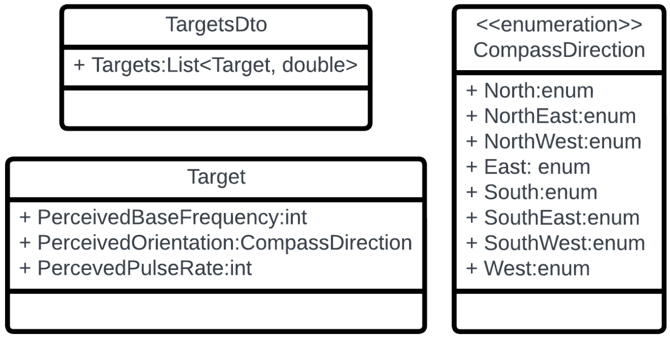

Figure 5.1: Designs for the information pertaining to communication of targets to other agents. The `double` value associated with the `Target` in the `Targets` list is the `TrackingConfidence`.

5.2.7 Sharing and inquiring traits designs

The study of the background theory revealed that for CPS to be effective in providing better solution, two important and intertwined factors that were of interest: willingness to ask for information, and willingness to consider shared information. Obviously, varying degrees of cognition received varying degrees of gains, such as *high cognition* subjects usually not relying too much on sharing, although most times when they did consider it they stood to gain from this concession. Meanwhile, another factor was the willingness to share information with others, which enabled weaker members of the community to eventually over-perform under the right condition (e.g. when shared by a *high cognition* subject with excellent problem-solving skills). Consequently, a combination of these traits could produce emergent phenomena that are bound to boost overall CPS performance.

The first two connected factors were thus combined into a single trait, which is labelled as `Inquisitive` and follows the established pattern of *high*/*low* values. This trait enables agents to determine how often they should ask for a state on targets from its peers, and how willing they are to consider the conveyed information. `Sharing` (*high*/*low*) is the other trait pertinent to CPS that was introduced to the agents, and it oversees how often they will be sharing targets information with their peers, to expand the perception of their peers to the larger world state so that they can make more informed decisions. The implementation of these traits is also as properties of the stateful inheriting class of the `IState` interface titled `SocialState`. Along with the `CognitiveState`, they live in the `IIntelligenceService` of the agent and factor into the decision-making process.

The agents can interact with each other using the following ways:

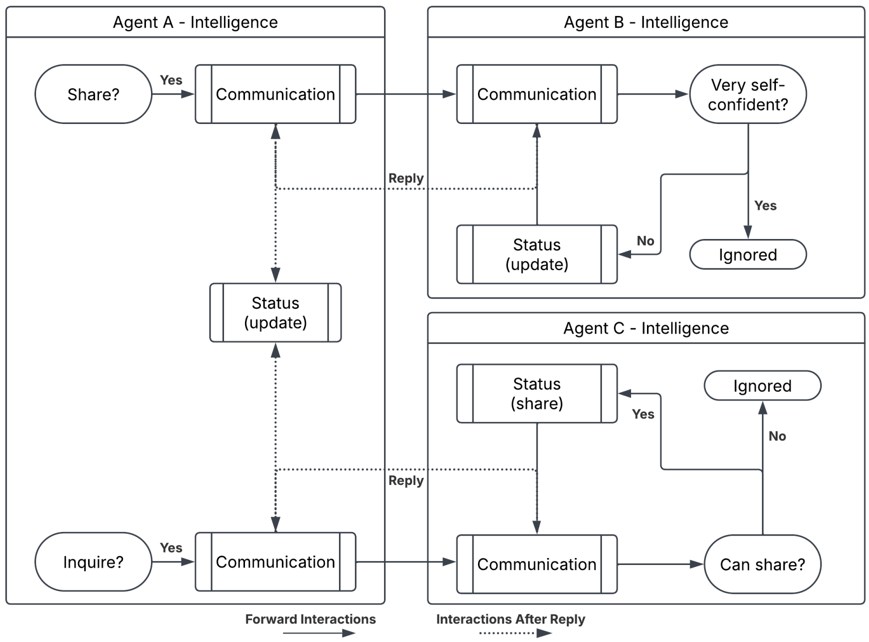

Figure 5.2: Microservice interaction designs demonstrating the socio-cognitive behaviours among agents.

High Inquisitive Will ask all their known peers.

Low Inquisitive Will ask all the most recently interacted with peer.

High Sharing Will share with all their known peers.

Low Inquisitive Will share with the most recently interacted with peer.

Most recently interacted with defines the peer that they last had an interaction with initiated by the agent itself, or interaction initiated by the peer themselves, whichever was the most recent case. Figure 5.2 presents the revised interactions among the intelligent agents when participating in socio-cognitive behaviours in this new CPS approach.

5.3 Final system implementation

The implementation state of the final system brought everything together to create the improved intelligent agents that will participate in the experiments in the more complex scenarios requiring CPS. Several parts of the system had to be changed at this point to tie everything: implement peer discovery for the agents, change the energy loss function (E_{loss}) to account for communication energy costs, and redefine the `IIntelligenceService` for the *perceive* step before acting to factor the new traits in the decision-making process. The design and architecture of the underlying distributed system framework for emergence allowed for ease of extension and introduction of new steps seamlessly, whether by re-using existing components and extending them in an appropriate manner or by introducing new steps to the agent life-cycle.

5.3.1 Agent peer discovery

The first important change in the final system was the introduction of peer discovery. The agents in this state of the system finally form an overlay network between them. For the purposes of the experiments at hand and this thesis, this has been determined as a Wireless Local-Area Network (WLAN). Borrowing from the Active Components framework capabilities exploited by the former work with EDBO [172], the current framework took it one step further to narrow down the WLAN IP address to a pre-set range that the agents will post REST requests for at the appropriate endpoint with the corresponding action: at `/` using `GET` request as described earlier for the `IDiscoveryService` implementation in Chapter 3.

When the agent receives a response to the request, the IP of the agent is stored in a newly implemented list within the agent object titled `KnownPeers` so that the agent can communicate with them as needed in the future (e.g., asking for target info, sharing target info). This is a task that happens the moment the agent is deployed in the environment and every $10th$ *stop-perceive-act* cycle. This ensures that the agent is always up-to-date and to simulate possible real-world applications where connectivity can be lost, or simply because a fellow tracker may have ran out of energy and has been disabled, thereby not attempting communication with non-operational agents and thus spending more energy than needed. The current implementation only covers a WLAN setting, but future endeavours could employ other protocols for the overlay network the agents will operate within as the scenario allows. Any agent that was last interacted with, even if discovered, is pushed to the top of the interaction preference list.

5.3.2 Energy loss refactoring

The second set of changes revolved around the new energy costs and thus the energy loss function described before has been expanded to factor in the new communication costs. One convenient fact about these new costs is that all types of communication in the system have the same processing overhead and manner of operation: a request with some low-sized data (no high transfer costs) is sent and a reply is received asynchronously when the other agent can reply. Together, this expected cost is akin to the cost of performing the listening task (E_l), albeit is not factored into the equation is the cost of keeping the WLAN system operating. It is assumed, at this point, that this is factored into the operating system energy costs and therefore only the required communication attempts are what incur the energy cost for evaluating agent behaviour and strategy efficiency. All actions pertinent to peer discovery, sharing target information, and inquiring about tracker information incur the same communication costs.

Consequently, the new energy loss function is:

$$E_{loss} = a \cdot E_m + b \cdot E_o + c \cdot E_l + d \cdot E_c = E_o \cdot \frac{320 \cdot a + 10 \cdot b + c + d}{10}$$

E_c The base energy cost for communication cycles.

d The total number of communication cycles (at least 1 at deployment).

5.3.3 Decision-making with traits

The last set of changes focused on redesigning the decision-making process for the agent. This process has been similar up until this point: "wake up" (stop), evaluate the vitals (perceive), decide based on the strategy (act). For the bio-inspired strategies, this *act* step was a simple choice between move and listen, while for *combined* it involved a more refined evaluation of the local state of the world known to the agent before deciding whether to move or listen. The fully trained *adaptive* strategy did not strain that far from this – only the training process did: ultimately it was a choice between a random action (on-policy) or the highest-rewarding one from the Q-table values (off-policy).

This particular and most important step in the *stop-perceive-act* cycle is still relevant for the socio-cognitive capable agents and will once more be a choice of move or listen, given that it will be using either the *combined* or the *adaptive* strategy. Inspired by the on-policy of Q-learning, which is the *ϵ-greedy* algorithm, the decision-making process introduces a step before acting that attempts to evaluate all socio-cognitive traits and determine if a sharing or inquiry communication attempt should take place before acting. RESTful API communication, barring the cases of network connectivity issues, happens in time negligible for the agent that usually takes only a few dozen milliseconds – average of $47.3ms$ through a test run of repeated 1000 calls while deployed for the current system. This means that an agent can send their call and if they do not receive an action for the defined $t_r = \Delta t - t_c$ (i.e. stop-perceive-act interval minus call await time), then they act as they normally would – otherwise they take that into account before action.

The traits and the tracking confidence weigh in on the outcome of the final decision, which is eventually a very simple outcome: keep following the same target or choose a new one from the suggested targets. When needed, a new target must be chosen among the suggested targets, because many replies can be received from more than one agent and thus a proper target must be selected. These targets are stored by the agent as they inquire of or receive from other trackers such information. Randomisation is included but not parametrised, to simulate more actual spontaneous behaviour just like *ϵ-greedy* does. To over-simplify the actual effect of the socio-cognitive trait management and social interactions: the agent may only change their desired target to track based on interactions and their traits. This is, as per the concept of emergence, a very small behaviour change (microscopic property) that could have a large impact on the solution to the problem in the long-term (macroscopic outcome).

In conclusion, the whole life-cycle of the agent there has evolved from the simple one presented in earlier chapters of this work. The final, abstract life-cycle is illustrated as a flowchart in Figure 5.3 to provide a better understanding of the complex *stop-perceive-act* life-cycle of the intelligent agents that attempt to solve the complex problem of energy-efficient tracking of mobile sound sources in highly dynamic acoustic scenes. This part of the work concludes **Objective A3-O1**, thereby introducing social behaviour to the system that can produce emergent phenomena, which can then be utilised towards improving CPS capabilities. This, however, remains to be determined through a final iteration of designing experiments and evaluating the results.

5.4 Preliminary Experiments with Emergence

5.4.1 Introduction

This section covers the familiar process of putting new strategies to the test following the same methodology as seen times and times over in the previous chapters and experiment sections, although the section regarding evolution of the system to accommodate the new requirements has already been detailed in the previous chapter. These experiments revolve around the research questions pertinent to the introduction of socio-cognitive traits to the system and undertaking the effort of attempting to solve a more complex problem this time, given that the trackers need to find all targets in the acoustic scene.

More importantly, this batch of preliminary experiments with the socio-cognitive traits aspire to determine if the introduction of those traits has had any effect on the expected system performance through emergent phenomena. Empirical evidence will be crucial in this case, lacking the tools to investigate in a more rigorous manner, and comparisons with CPS performance baseline regarding high/low socio-cognitive trait research, as well as the existing results from previous experiments. This is also the final round of experiments in this thesis, and therefore the ultimate chance to get more information on the performance indicator, in addition to the best possible strategy for the problem at hand and especially in more complex acoustic environments and tracking goal requirements (e.g. to locate all targets).

5.4.2 Research questions

With the aim towards the conclusion of this thesis, there remain a few important research questions that ought to be answered, in addition to a couple of questions that were posed in previous sections and that are now relevant for the brand new and more intricate approach utilised towards solving the problem. The cases being tested for performance included tracking all targets, something which was

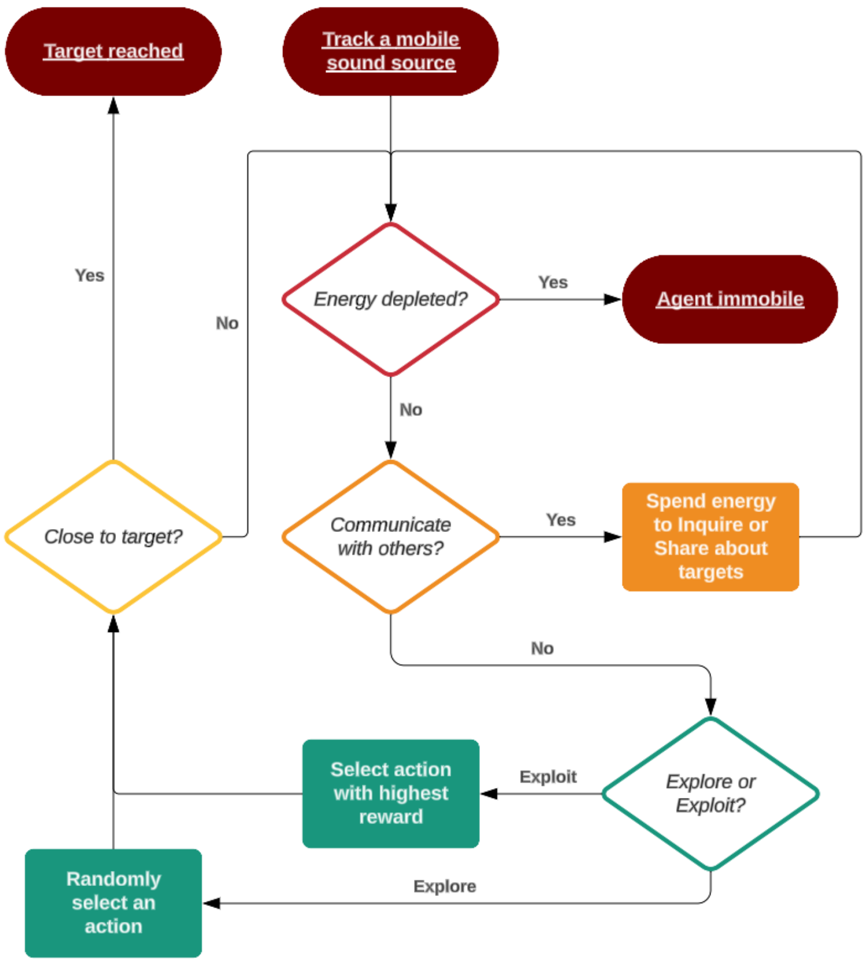

Figure 5.3: The final agent life-cycle of the *adaptive* strategy now infused with socio-cognitive traits and new behaviours, illustrated at a high abstraction level in this flowchart.

done in past experiments with benchmarks to compare to, raising the question of how much performance does the newly introduced socio-cognitive stand to gain from this. This is an important question to gauge how much energy and time efficiency is gained at the cost of the newly introduced communication costs, leading to a better performing system for more complex cases where CPS is essential.

One more interesting question that can be answered with the new system is how the findings from socio-cognitive CPS studies align or not with the behaviour displayed by the intelligent agents, when the analogous socio-cognitive characteristics are introduced to them in the microscopic level. An answer to this question, whether they match closely or not, can have a larger impact on future studies related to emergence in CASA applications and problem-solving. If such microscopic-level properties can produce desirable traits in the macroscopic level in a system comprised of AI entities working towards solving a complex problem, then further research in related bio-inspired fields can offer similar benefits to other real-world problems and applications.

Furthermore, emergent phenomena, if any can be observed and identified empirically through the experiments, forms another interesting research question pertinent to the problem at hand. Naturally, a more rigorous and disciplined framework employing modern machine-learning methods for identifying such phenomena would give more definitive and credible answers. Nonetheless, this attempt can explore this direction empirically through the experience with previous related work, as well as with experience gained within the confines of the study at hand, too.

In conclusion, the research questions for the preliminary study in emergence are:

RQ1 How do the different socio-cognitive combinations perform when tracking *two* targets in an environment with *few* obstacles?

RQ2 How do the different socio-cognitive combinations perform when tracking *two* targets in an environment with *many* obstacles?

RQ3 How do the different socio-cognitive combinations perform when tracking *two* targets in a *highly dynamic environment*?

RQ4 Is the expected contribution of socio-cognitive traits reflected in the performance of the intelligent agents?

RQ5 Have the socio-cognitive traits induced emergent behaviour in the distributed system?

5.4.3 Experiment designs

The last experimentation phase in this thesis faced a serious problem with the demands pertaining to the number of experiments sets that would be required to have a 100% coverage of all possible combinations of socio-cognitive traits thrown in an assortment of scenarios pertinent to the research questions being posed. This resulted in the creation of a long list of required experiment sets that ought to be considered and a need to reduce that number in a meaningful manner so that the research questions can be answered adequately through this process. To begin with, nonetheless, it is essential to review the possible sets.

Firstly, there are 4 different cognitive trait combinations (2 traits with 2 possible values each) that were associated with 4 different social trait combinations (same as for cognitive) for a total of 16 socio-cognitive trait combinations for each AI component. Moreover, these would need to be combined with each other in scenarios where the two trackers were not the exact same in nature, as for example not being both with *high* values in every trait but one with all *high* working together with one of all *low* etc. Evidently, a combination of 256 unique trait combinations for two trackers exist, each to be tested in 3 different scenarios: *low* and *high* obstacle density environments, as well as the highly dynamic environment. Consequently, the unyieldingly large number of $768,000$ experiments ($3 \cdot 256$ sets of $1,000$ experiments each) had to be averted.

The core questions **RQ1-RQ3**, are questions that need to be answered specifically for the *adaptive* strategy, which is the more performant of the strategies and the increased efficiency and efficacy in it are the most desired outcomes of this endeavour. This suggests a total of 8 combinations of traits, provided that the `ProblemSolving` remains constant in this case, with 3 types of experiments for a total of 24 sets of experiments. **RQ4** can be answered with some of these results, but there is merit in seeking to determine whether these traits had any effect in the less capable strategies or not, which is also something that is expected to produce results towards answering **RQ5**.

Regarding **RQ5**, more dramatic emergent behaviour is expected to emerge from the less competent strategies, and through the interaction of the higher socio-cognitive agents with the higher social but lower cognitive capabilities agents. This expectation allows for a significant reduction in the sheer number of combinations for experiment sets required to elicit answers. Accordingly, additional experiment sets were added that cover the cases of a variance between *low* and *high* social traits in agents with static *low* cognition alone (i.e. *combined* strategy and *low* self-confidence), in all 3 scenarios of interest: highly dynamic environment, low, and high obstacle density environments. In closing, 12 more experiment sets when the *low* cognition experiment sets are added for a grand total of 36 experiment sets.

5.4.4 Results

For the gathering and analysis of the results, the familiar (T, E, S, O, M) tuple defined at the start has been utilised again and extended only so far as that the S part has been revised to also display now the strategy along with the current values of socio-cognitive traits. This specific iteration of experiments has been the more extensive in numbers, although thanks to the existing implement mechanisms for analysing the results and capable of accommodating the new requirements seamlessly, the process was performed efficiently. The research questions were revisited with the results, which are gathered in Table 5.1 for a

	Success Rate (%)	**Time-to-target (*s*)**	**Energy Left (%)**
sis-2F	95	21	79
siS-2F	95	21	78
sIs-2F	96	23	76
sIS-2F	95	22	77
Sis-2F	93	21	80
SiS-2F	93	20	81
SIs-2F	94	22	79
SIS-2F	92	20	83
sis-2M	94	21	85
siS-2M	93	19	86
sIs-2M	94	24	82
sIS-2M	92	20	81
Sis-2M	91	19	84
SiS-2M	93	19	83
SIs-2M	92	22	85
SIS-2M	92	18	83
sis-HD	90	21	66
siS-HD	89	20	68
sIs-HD	93	23	62
sIS-HD	91	21	64
Sis-HD	90	19	69
SiS-HD	91	18	70
SIs-HD	93	22	66
SIS-HD	92	20	67
Cis-2F	88	8	90
CiS-2F	87	8	91
CIs-2F	92	9	88
CIS-2F	90	8	90
Cis-2M	88	8	91
CiS-2M	89	9	92
CIs-2M	92	8	89
CIS-2M	92	9	90
Cis-HD	69	5	99
CiS-HD	68	5	101
CIs-HD	71	6	97
CIS-HD	69	6	100

Table 5.1: Success rates, time-to-target, and energy remaining rounded to closest integer for the *socio-cognitive* experiment runs. Lowercase letters **i** and **s** represent *low* `Inquisitive` and `Sharing` traits respectively, and uppercase *high* values accordingly. Lowercase **s** letter represents *low* `SelfConfidence` experiments for *adaptive* strategy, whereas uppercase is for *high* value trait. **C** represents the experiments for the *combined* experiments. **2*** and **HD** keep representing the familiar dual-tracker (one *adaptive* one *combined*) with varied obstacles and highly dynamic environments respectively.

concise representation and need to be consulted for the results presented next, once more categorised by energy-time efficiency and success rate.

Beginning with the latter, the **success rates** of the experiments involving only the *adaptive* strategy showed only a minor increase within the range of an additional $2-5\%$ for all experiment scenarios, slightly boosting the already impressive success rates of the strategy. The more significant gains, though, come in the case of the *combined* strategy (i.e. the lower cognition experiments). The success rates for this strategy with the *low* social trait values have demonstrated an increase of $+5.5\%$, while an even more substantial average difference of $+9\%$ is displayed for *high* `Inquisitive` trait for the dual tracker experiments of both few and many obstacles over the *low* value. On the other hand, the increase does not appear to be as substantial on average ($+3.3\%$) for any cases involving the highly dynamic scenario.

Energy efficiency for the *adaptive* strategy only experiments has shown an improvement across the board, too, and in this case contrary to the success rates analysis the increase is much more impactful. Specifically, of high interest is the example of *high* `Inquisitive` and *low* `Sharing` demonstrating an increase of $+7\%$, which is evidently almost double the increase of $+4\%$ shown for *both high* social traits, when value of `SelfConfidence` cognitive trait is *low* for the important highly dynamic environments. Meanwhile, the *high* `SelfConfidence` trait does not appear to gain much in final energy reserves as compared to the *low* values, whilst a look at higher `Sharing` traits do not appear to offer any groundbreaking energy efficiency gains, either. *Combined* strategy showed similar low gains for the less dynamic environments, albeit it did actual show a very important loss (-9%) with regards to the highly dynamic environment.

In closing, the **time efficiency** showed an almost analogous increase to energy efficiency (i.e. decrease in the time-to-target T), when comparing Tables 5.1 and 4.5, in relation to the *adaptive* strategy. Indeed, even the variance of gains between the two are within the margin of $\pm 2\%$ (in seconds). As such, these results on time should be discussed in the context of the results to energy, too, for a potential empirical relationship. The *combined* strategy did verify the low analogous increase for less dynamic and decrease for more dynamic environment scenarios as with energy, also, nevertheless the gains in success rates are substantial enough to merit the experiments with socio-cognitive traits.

5.4.5 Discussion

The first set of research questions, **RQ1-RQ3**, were designed with the evaluation of the performance of the *adaptive* strategy in mind, and with whether new micro-properties could produce macro-benefits to CPS scenarios. The increase of efficiency across all of the tree metrics – energy, time, and success – is the answer to these questions, illustrated in Figures 5.4 and 5.5. Indeed, the introduction of socio-cognitive traits to the agents had a beneficial effect and the strategy. Even with the introduction of these basic traits alone, it has been enough to increase the performance of CPS when attempting to locate all targets in any scenario. Notwithstanding, there were also important information

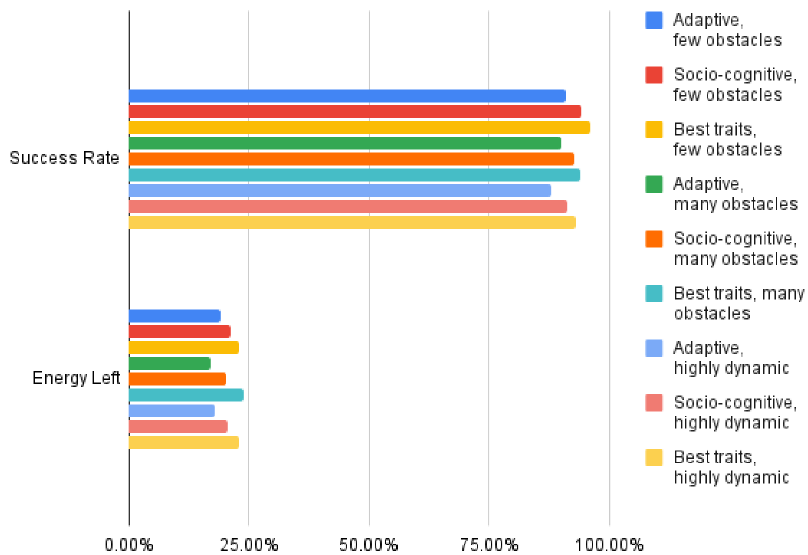

Figure 5.4: A graph representing the metric percentages for the strategies using *adaptive vs. socio-cognitive* (average for all trait combinations) vs. the most performant socio-cognitive trait combination (low confidence, highly inquisitive) in the important cases they were tested in. Highlights overall performance gains for the evolution of the strategy for each case.

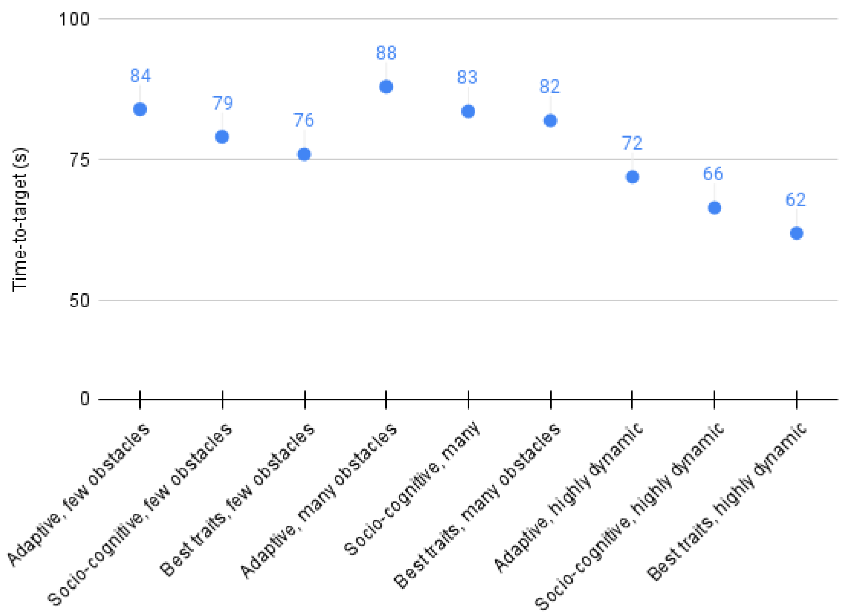

Figure 5.5: A scatter graph representing the time-to-target for the strategies using *adaptive* vs. *socio-cognitive* (average for all trait combinations) vs. the most performant socio-cognitive trait combination (low confidence, highly inquisitive) in the important cases they were tested in.

to derive with regards to the extent that such traits affected performance and the implications these pose, also in the context of emergence.

Particularly, the first set of interesting observations pertains to the `SelfConfidence` trait of agents, and how if it is *high*, it does not offer that much to the capabilities of the strategy in success rates and efficiency. For the success rates it is inarguably the result of the fact that the strategy is already very capable at succeeding in tracking both targets in a timely manner (i.e. *high* `ProblemSolving` representing the adaptive strategy), even so when it comes to energy and time efficiency there are less gains due to the fact that the strategy chooses to ignore the input about trackers of others due to the high confidence. Notably, when the co-tracker possesses high `Inquisitive` behaviour there is a somewhat higher gain instead, arguably due to increased suggestion inputs triggering more frequently the algorithm to choose a suggestion and correct the course and gain time in finding the extra target.

Low confidence, on the other hand, enjoyed bolstered efficiency in all cases, suggesting that communication between the cooperative agents in a CPS scenario is of paramount important. How this relates to the social behaviour is through the link between a *low* `SelfConfidence` prompting more interactions with getting better tracking suggestions from partners, but it also appears to be reliant on the `Inquisitive` trait being *high* as well, thereby urging the agent to consider the information they retrieve. These results do adhere to the findings of the pertinent social studies explored earlier [211, 213, 212].

Moreover, this could also form an empirical relationship between the increased efficiency when meaningful collaboration takes place, such as the case of a competent tracker suggesting to another competent tracker that they can indeed pursue another target instead as the former is confident to succeed and the latter can make up for time and energy spent. This is an emergent phenomenon where *division of workload* and *delegation* take place, as the trackers decide between them which target each one should track to better achieve their goal, giving a partial answer to **RQ5**.

Ultimately, the results for the already expert *adaptive* tracking strategy can be boiled down to the following observation: when there is a mix of a high-confidence, high-social tracker joining forces with a low-confidence, highly inquisitive tracker. This alludes to the scenario where the confidence of the more capable overall target allows them to play the role of a potential coordinator in the community of agents in a CPS. This emergence of a potential leader in the community is one more emergent phenomenon, the emergence of leadership and fellowship such studies suggest, initial signs of which can be observed empirically through these results.

This is evidenced by the higher gains and better success rates and efficiency being found in this specific collaboration scenario. In fact, this is more pronounced in the highly dynamic environment than the less dynamic ones, albeit the gains in the latter category are still of utmost importance, another significant contribution towards the answer to **RQ5**: through the socio-cognitive traits there emerged leaders, fellowships, workload division, and delegation. Lastly, this prompts potential for research with a higher number of collaborating agents to establish more rigorously how these relationships and statuses emerge, as well as if they would be able to handle the workload division and delegation with similar efficiency and efficacy.

Finally, the results did not produce satisfying results with regards to the *low* `ProblemSolving` strategy. Indeed, the higher success rates were welcome, but there were no gains – and in fact in some observations losses instead – when the important costs in energy and time were to be considered. This also inarguably suggests that a capable tracking strategy trained for such applications is required first to improve it with socio-cognitive micro-properties: the time lost due to subpar obstacle tackling and open-space recognition cannot be overcome by these micro-properties in the macroscopic level. This provides a final answer to the **RQ4** question, with a caveat: yes, these traits improve the tracking efficiency behaviour of intelligent agents as the social studies suggest, albeit only in the context of already efficiently developed solo tracking strategies. With these final answers, **Objective A3-O2** has been achieved, concluding the objectives for this part of the work.

5.5 Conclusions

The core focus of the work in this chapter centred around improving the problem-solving capabilities of both the developed framework and the developed tracking strategy under CPS circumstances, the core contributions previous presented in this thesis. This chapter concludes the work regarding the attempted improvements of an already viable and performant tracking strategy inspired by emergence and pertinent studies from the domain of the social sciences. The evolution of the strategies passing through the four consecutive steps of *bio-inspired* to *combined* to *adaptive* to infused with *socio-cognitive* traits has been documented extensively. The three apropos efficiency metrics (i.e. success rates, energy left, and time-to-target) have been improved over the course of this thesis for the important dynamic acoustic scenes the solution has been developed for.

The work in this chapter illustrated the worthiness in seeking for inspiration in fields that can contribute to emergence in intelligent agent interactions when seeking solutions to complex problems through distributed systems utilising a cooperating approach. The social studies relevant to CPS complemented expertly, even with their elementary implementation, the *adaptive* strategy in provided and even more optimal solution, whilst also showcasing the potential to improve even in specific and reduced capacity even less performant options. Results were in sync with the observations detailed in the cases applicable to humans and validated through the experiments for the AI nodes exhibiting such behaviours.

Results showcased the capacity of complex emergent phenomena to appear from the simple introduction of even the most basic new microscopic properties. Indeed, in the macroscopic level, hints towards the emergence of leadership in the community of collaborating agents were found. Additionally, meaningful collaboration was facilitated through sharing of pertinent information in a truthful manner (i.e. with the accompanying confidence level for own tracking capabilities). Any other emergent phenomena were not observed due to reliance on empirical observations, albeit more could be possible through a more disciplined approach and extended experimentation. Both contributed to improving the performance of the end-system and provide more parameters to fine-tune in any future attempts at improving the end-system.

Regrettably, a limitation of this thesis was the number of possible experiments sets that could be run within the allotted time-frame due to the given exponential increase of required experiments even with a simple new factor and new research questions. A new social or cognitive trait, or even something artificial inspired by studies on emergence contributing to self-* properties and CPS cases. What new emergent phenomena could be possible when introducing one more tracker and target, or is the leadership expectations validated with proper experiment setups? More fine-grained levels of behaviour assigned to the traits could also provide a better performance, such as a *medium* level of self-confidence, which could not be covered through this thesis.

In conclusion, this chapter offered two major contributions towards realising the final goal of this thesis, **Goal A3**: it showcased both a methodology for implementation and results that can be utilised, even out-of-the box as a solution for a real-world application with the necessary tweaking (e.g. proper energy-to-battery-to-actions mapping for the strategies). Future work opportunities arise for even more interesting avenues to follow and exploit in providing more optimal solutions. With this part of the work concluded, the overarching study for energy-efficient CASA tracking of mobile targets has reached its end.

Chapter 6

Conclusions

6.1 Key Outcomes

This thesis started with a heavy focus on interdisciplinary research, inspired by previous work with bio-inspired computing and distributed systems, having contemplated the potential to apply such knowledge in the field of CASA to solve complex problems in a novel manner. Through this preliminary dive into pertinent research on the domains of interest, the gap in research endeavours with potentially impactful applications became clear: virtually non-existent energy efficiency management in audio tracking solutions, lack of focus on the capabilities of a mobile listener, or even oversight in exploiting the astounding behaviours animals exhibit related to listening and tracking. Consequently, a plan was laid out to follow a methodology for combing everything in a single complex problem that could have real-world applications in assorted fields, such as target tracking in crowded and noisy environments, or locating targets under threat in disaster scenarios were devices operating on listening alone can offer solutions efficiently.

This undertaking aspired to leave behind the following legacy: a framework for developing CASA applications to solve even more complex problems in the future, a methodology in a stepwise manner towards borrowing themes from various domains and evolving them into real and novel CASA solutions, thus ultimately providing a solution for a well-defined problem where none were found to exist. Accordingly, a distributed systems framework for CASA applications has been developed to be used with concepts inspired by biology and established AI techniques, paving the way towards the formulation of an energy-efficient tracking strategy for highly dynamic acoustic scenes, and lastly leveraging ideas from the social sciences towards beneficial emergent phenomena and CPS to develop an even more efficient cooperation model for such scenarios.

6.1.1 A distributed system framework for CASA

Developing such a CASA distributed system framework required studying the state-of-the-art in distributed systems, sensor networks, and related solutions, whilst combining them with the most modern, efficient, and future-proof development tools to overcome reliance on outdated and low-performance solutions. The system was designed to be highly extensible by providing a basis that required minimal implementation to build upon and create an application, as well as for running simulations. Key feature is the ease of switching from simulation to real-world application. Capacity for having a robust, scalable, and flexible in environments wrought with uncertainty even for disaster scenario applications.

Moreover, the system demonstrated the capability of acting as a framework for future solutions by accommodating an assortment of AI techniques for intelligence and communication over the course of the study: from bio-inspired algorithms to artificial ones for reactive agents, to machine-learning for both training and execution, and even to directly interacting and cooperating nodes by virtue of minimal parametrisation. Simulation performance was fast, albeit not as fast as simulation-only frameworks due to the overhead of calling costly functions at both the operating system and higher logic layers. Balancing between real-world and simulation performance is another key feature of this framework. The limitation of factoring in the experiments real-device operating costs could be studied through future work, lending further credit to the lightweight implementation offered by the end-system when deployed in real-world applications.

6.1.2 An energy-efficient tracking strategy

A major outcome of the study is the developed *adaptive* strategy, which was the lengthier part of the process in the lifetime of this work. This process of developing the strategy delivered a robust tool for performing the tracking of mobile audio sources in both high and low dynamic acoustic scenes efficiently with regards to both time and energy spent. Notably, there are specific assumptions about the environment that were required to facilitate these developments, such as the type of room and acoustic characteristics of the scene, that could potentially narrow down its implementation in different scenarios. Nevertheless, the background theory and means for adapting this approach to other environments now exists by virtue of this thesis, but also for repeating pertinent experiments in such new environments to ascertain its applicability without modification.

At the same time, the process of developing the *adaptive* strategy offered a methodology for starting from something simpler and even borrowed from external fields of research and reaching this point. This demonstrates a path towards achieving desired results in bio-inspired computing through an iteration of modelling, experimentation, and exploitation of outcomes with consistency: start from a simple inspiration from biology and something with potential forming a baseline, attempt to combine the best parts of it in an artificial attempt to improve upon the baseline, and finally how to employ advanced tools for optimal control solutions to attain higher efficiency and efficacy.

While the approach has been established for several related research endeavours and software design over the years, thus by no means claiming innovation, this one demonstrates its disciplined application to the field of optimisation in the context of CASA when inspiration is elicited from animal behaviours. Indeed, the results by the end of this process highlighted the significance of not skipping a step in this process and rushing head-first towards relying on the trending machine-learning paradigms alone. Such arguments for this are the explosive breeding movement patterns and the complementarity showcased by the two different bio-inspired strategies – even if in the end the *adaptive* strategy outperformed them, it allowed for combining different strategies in the emergence experiments that followed (i.e. different traits, better overall system performance).

Invoking the effect of explosive breeding in arboreal locomotion inspiring various degrees of movement, or the frequency of movement in general, both of which could have been missed otherwise, is indicative of what significant profits can be reaped by lending more time to this stepwise design and implementation instead at the cost of more time. This trade-off between time and possible different optimisation approaches not being attainable or having to be discounted, has naturally constituted an impediment in potentially achieving more optimal goals throughout the study. These can be ameliorated via future work now that the core exists, but it also outlined the weight of choosing the right tool for the work given such constraints (i.e. Q-learning instead of SL and similar). Ultimately, the results of the strategy as it has evolved can also stand as a measure for future work in the field with similar aspirations – something missing as the domain research highlighted.

6.1.3 Emergent solutions in cooperative tracking

The final contribution of this thesis brought another dimension into the bio-inspired optimisation design and implementation: that of social sciences and what communities can achieve when working together towards solving a goal. Moving away from animal behaviour, which excels in providing unique solutions often overseen by humans, and going towards the behaviour of the latter, which has multi-layered and much more elaborate communication and interaction mechanisms, is an indication of what emergence can offer when explored diligently. Admittedly, it also brought to attention the vast space it can occupy and the demands for design, experimentation, and evaluation it can evoke, such as four simple traits causing an exponential demand in simulations, more cumbersome analysis, and far more intricate architecture considerations.

Despite the challenges posed by the workings of emergence, narrowing the focus down to the bare necessities that could be identified as beneficial to the complex problem at hand, it did become possible to introduce the important microscopic properties to the intelligent agents that eventually provided significant macroscopic profits through basic emergent phenomena. The already performant strategy evolved into a more efficient form when in cooperating tasks, especially so under certain conditions. Some agents can act as coordinators or leaders, depending on their adopted socio-cognitive traits, and others as more reliant on their betters to perform a task that improves overall system efficiency.

Even with these simple additions, this can inarguably serve to suggest the potential benefits that can be leveraged from introducing more such properties, running the experiments with many more actors, and potentially with a more rigorous analytical framework of for determining when emergence occurs or to which degree it should be allowed in the system. Despite such future considerations, the final result of this thesis is clear: an *adaptive* strategy that can perform efficiently in an isolated manner, and a way to improve its high performance even further when more tools are available, or the scenario is more complex.

6.2 Future Work

Reflecting on the progress of this work, the results suggest that it manages to provide a robust solution to the energy-efficient tracking of mobile audio sources in dynamic environments with smart devices, whether they need to perform the task alone or in tandem with other devices operating in an overlay network. Apart from the strategies, the distributed system framework tailored for CASA applications has so many more possibilities to offer for future work in this specific field, and potentially to more problems if desired and extended in the proper manner demonstrated in latter steps. Up until this point, when discussing the results of evaluations and when designs were presented, a wealth of such opportunities has been revealed, which this section aspires to organise in a classified manner.

The limitations of this thesis have been highlighted at each step of the way, being one of the first attempts to improving this work and making the end-product more robust. Meanwhile, optimisation of the training process using more steps or better designs, or even different AI techniques and approaches, can offer significant results. Lastly, the numerous avenues that open up thanks to emergence by building on the provided baseline can offer boundless research opportunities, whether for purely analytical, scientific purposes or for focused strategy optimisation undertakings. These will be organised in the following paragraphs in the form of research questions asked and, briefly, how they can be addressed, as well as what they can offer, using the fundamental tools created through this work.

How does the system perform when deployed on real devices? Knowing that the system can be deployed on assorted smart and robotic devices by virtue of its development tools, a new baseline can be established by measuring the performance of the *adaptive* strategy as is. It can also provide proper evaluation for the assumptions made and indicate whether this is a required step for an optimal result. Implementation can be achieved by simply replacing the energy cost reductions after each logical action by instead changing the function to call the interface of the motors of the device.

What performance costs do the software requirements incur? This work could complemented the above work by offering even better measurements for energy costs and actually developing a "personalised" strategy for specific families of devices, but it can also stand alone to offer an approximation strategy for energy costs in related studies. Actualisation of this can be done by deploying the system on an real device and measuring for the expected operational time-frames the costs when running the system (e.g., operating system, Wi-Fi networking, motor costs).

Can the *adaptive* strategy achieve higher convergence? Due to the time constraints and the scope of this project, hyperparameter tuning and convergence estimation have not been as diligent as they could potentially have been. Such work can help alleviate this issue and, if the answer is yes, result in an even stronger strategy. This would require a revised training approach, more fine-grained control of hyperparameters, far increased training episodes, supported by a similar convergence metric combining BFS and RMSE but reviewed in every single step.

Do revised energy costs train a significantly different strategy? Pertinent to the first – and possibly combined with the second – question above, this one can help ascertain whether such changes can cause smaller or larger changes to the eventual optimal policy after training. Utilising a revised energy cost policy and putting the strategy through the same training can provide an answer to this question. Possible new observations on how the agent learns any new things, that is if it really changes that much, will also be valuable.

Does *X* outperform Q-learning for this specific problem? This question would require much more extensive work to do so compared to the previous questions. This X could range from using continuous -space RL algorithms to elaborate techniques such as neural networks or even extending the Q-learning implementation to become Deep RL. The feasibility of modelling the dynamic environment is unknown, as is the feasibility of developing a framework for producing input-output pairs for SL attempts, therefore this can be a risky endeavour. The distributed framework offers the tools to implement it, nonetheless, while examples of adapting it to the needs of such a project can be found in the various sections detailing system changes occurred by each strategy evolution phase.

Does the new trait *X* offer gains to the strategy in CPS? This is a question that could require expanded research, and even across domains if not within a single domain, to determine what trait X to introduce. Thankfully, the framework and the existing work from this thesis outline the steps to take after that and how to carry out this process in an effective manner. A cautionary sign is raised here, so that the trait is rigorously evaluated with all other traits and possible combinations, albeit at the cost of time, as it could offer benefits in specific cases of interest, still becoming a detriment in others.

Are there more emergent phenomena to be leveraged? While this question is less clear than all before it, it is so that it can be rephrased by specifying how one expects to approach it: better emergence identification mechanisms (e.g. by machine-learning), higher volume of experiments (e.g. by focusing on skipped trait combinations), expanded number of actors (e.g. bigger communities and more potential forms of interactions and outcomes). The latter is the one suggested to begin with, as it is evidently the biggest limitation of the chapter focusing on emergence in this work. Despite requiring a mere addition of more entities to the environment layer of the framework and running experiments, the sheer number of them required were deemed infeasible for the scope and time allotted to this thesis – thereby the preliminary results having to suffice.

Bibliography

[1] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications.* Wiley-IEEE press, 2006.

[2] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*, 1994.

[3] W. A. Yost, "The cocktail party problem: Forty years later," in *Binaural and spatial hearing in real and virtual environments*, 1997, pp. 329–348.

[4] H. Witte, H. Preuschoft, and M. S. Fischer, "The importance of the evolutionary heritage of locomotion on flat ground in small mammals for the development of arboreality," *Zeitschrift für morphologie und anthropologie*, pp. 221–233, 2002.

[5] J. A. Doherty and H. C. Gerhardt, "Acoustic communication in hybrid treefrogs: sound production by males and selective phonotaxis by females," *Journal of Comparative Physiology A*, vol. 154, no. 3, pp. 319–330, 1984.

[6] B. C. J. Moore, "Basic auditory processes involved in the analysis of speech sounds," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 363, no. 1493, pp. 947–963, 2008.

[7] C. Kung, "A possible unifying principle for mechanosensation," *Nature*, vol. 436, no. 7051, pp. 647–654, 2005.

[8] A. J. Hudspeth, "How the ear's works work," *Nature*, vol. 341, no. 6241, pp. 397–404, 1989.

[9] A. Hudspeth, *Making an Effort to Listen: Mechanical Amplification in the Ear*, 2008, vol. 59, no. 4.

[10] A. J. Hudspeth and M. Konishi, "Auditory neuroscience: development, transduction, and integration." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 22, pp. 11 690–1, 2000. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=34336&tool=pmcentrez&rendertype=abstra

[11] H. Hudde and C. Weistenhofer, "Key features of the human middle ear." *ORL; journal for oto-rhino-laryngology and its related specialties*, vol. 68, no. 6, pp. 324–8, 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17065824

[12] J. D. Bronzino, *Biomedical Engineering Fundamentals (The Biomedical Engineering Handbook, Third Edition)*, 2006.

[13] W. M. Roberts, J. Howard, and a. J. Hudspeth, "Hair cells: transduction, tuning, and transmission in the inner ear." pp. 63–92, 1988.

[14] W. A. Yost, "Audition." in *Handb. Psychol. Exp. Psychol. Vol. 4*, 2003, pp. 121–146. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2003-04681-005&site=ehost-live http://search.ebscohost.com/login.aspx?direct=true%7B&%7Ddb=psyh%7B& 04681-005%7B&%7Dsite=ehost-live http://search.ebscohost.com/login.aspx?direct=true%25

[15] T. M. Talavage, M. I. Sereno, J. R. Melcher, P. J. Ledden, B. R. Rosen, and A. M. Dale, "Tonotopic organization in human auditory cortex revealed by progressions of frequency sensitivity." *Journal of neurophysiology*, vol. 91, no. 3, pp. 1282–1296, 2004.

[16] J. S. Snyder, M. K. Gregg, D. M. Weintraub, and C. Alain, "Attention, awareness, and the perception of auditory scenes," *Frontiers in psychology*, vol. 3, p. 15, 2012.

[17] J. H. Kaas and T. A. Hackett, "'what'and'where'processing in auditory cortex," *Nature neuroscience*, vol. 2, no. 12, pp. 1045–1047, 1999.

[18] R. J. Zatorre and P. Belin, "Spectral and temporal processing in human auditory cortex," *Cerebral cortex*, vol. 11, no. 10, pp. 946–953, 2001.

[19] E. Hemery and J.-J. Aucouturier, "One hundred ways to process time, frequency, rate and scale in the central auditory system: a pattern-recognition meta-analysis," *Frontiers in computational neuroscience*, vol. 9, p. 80, 2015.

[20] R. Näätänen and I. Winkler, "The concept of auditory stimulus representation in cognitive neuroscience." *Psychological bulletin*, vol. 125, no. 6, p. 826, 1999.

[21] S. E. Palmer, "Vision science," *System*, p. 810, 1999.

[22] L. P. A. S. van Noorden, "Temporal coherence in the perception of tone sequences," 1975.

[23] W. M. Hartmann, "Pitch perception and the segregation and integration of auditory entities," *Auditory Function, Neurobiological Bases of Hearing*, pp. 623–645, 1988.

[24] C. J. Darwin, "Auditory grouping," *Trends in cognitive sciences*, vol. 1, no. 9, pp. 327–333, 1997.

[25] M. R. Jones and W. Yee, "Attending to auditory events: the role of temporal organization," in *Thinking in Sound The Cognitive Psychology of Human Audition*, 1993, pp. 69–112.

[26] B. C. J. Moore, "An Introduction to the Psychology of Hearing," *Boston Academic Press*, vol. 3, p. 413, 2003.

[27] C. J. Darwin, R. W. Hukin, and B. Y. Al-Khatib, "Grouping in pitch perception: evidence for sequential constraints," *Journal of the Acoustical Society of America*, vol. 98, no. 2, pp. 880–885, 1995.

[28] C. Alain, S. R. Arnott, and T. W. Picton, "Bottom–up and top–down influences on auditory scene analysis: Evidence from event-related brain potentials." *Journal of experimental psychology: human perception and performance*, vol. 27, no. 5, p. 1072, 2001.

[29] L. Rosenblum, "Audiovisual speech perception and the mcgurk effect," *Oxford Research Encyclopedia, Linguistics*, 2019.

[30] N. Itatani and G. M. Klump, "Animal models for auditory streaming," *Phil. Trans. R. Soc. B*, vol. 372, no. 1714, p. 20160112, 2017.

[31] T. Aubin and P. Jouventin, "Cocktail-party effect in king penguin colonies," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 265, no. 1406, pp. 1665–1673, 1998. [Online]. Available: http://rspb.royalsocietypublishing.org/content/265/1406/1665

[32] T. Lengagne, T. Aubin, J. Lauga, and P. Jouventin, "How do king penguins (Aptenodytes patagonicus apply the mathematical theory of information to communicate in windy conditions?" *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 266, no. 1429, pp. 1623–1628, 1999. [Online]. Available: http://rspb.royalsocietypublishing.org/content/266/1429/1623

[33] C. F. Moss and A. Surlykke, "Auditory scene analysis by echolocation in bats," *The Journal of the Acoustical Society of America*, vol. 110, no. 4, pp. 2207–2226, 2001.

[34] M. Lewicki, B. Olshausen, A. Surlykke, and C. Moss, "Scene analysis in the natural environment," *Frontiers in Psychology*, vol. 5, p. 199, 2014. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2014.00199

[35] M. A. Bee, *Treefrogs as animal models for research on auditory scene analysis and the cocktail party problem.*, 2015, vol. 95, no. 2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167876014000051

[36] A. S. Bregman, "Progress in understanding auditory scene analysis," *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 1, pp. 12–19, 2015.

[37] M. A. Bee and C. Micheyl, "The "Cocktail Party Problem"; What Is It? How can It Be Solved? And Why Should Animal Behaviorist Study it?" *Jurnal Comparative Psichology*, vol. 122, no. 3, pp. 235–251, 2009.

[38] M. A. Bee and J. Christensen-Dalsgaard, "Sound source localization and segregation with internally coupled ears: the treefrog model," *Biological Cybernetics*, vol. 110, no. 4-5, pp. 271–290, 2016.

[39] H. C. Gerhardt, "Female mate choice in treefrogs: Static and dynamic properties," *Animal Behaviour*, no. 42, pp. 615–635, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003347205802453

[40] A. M. Simmons, R. C. Buxbaum, and M. P. Mirin, "Perception of complex sounds by the green treefrog, Hyla cinerea: envelope and fine-structure cues," *Journal of Comparative Physiology A*, vol. 173, no. 3, pp. 321–327, 1993.

[41] S. H. Hulse, "Auditory scene analysis in animal communication," *Advances in the Study of Behavior*, vol. 31, pp. 163–200, 2002.

[42] K. D. Wells, "The social behaviour of anuran amphibians," *Animal Behaviour*, vol. 25, no. PART 3, pp. 666–693, 1977.

[43] D. B. Kelley, "Vocal communication in frogs," *Current Opinion in Neurobiology*, vol. 14, no. 6, pp. 751–757, 2004.

[44] A. R. Lammers and U. Zurcher, "Stability during arboreal locomotion," *Theoretical biomechanics*, pp. 319–334, 2011.

[45] H. Preuschoft, "What does" arboreal locomotion" mean exactly and what are the relationships between" climbing", environment and morphology?" *Zeitschrift für Morphologie und Anthropologie*, pp. 171–188, 2002.

[46] F. J. Zamora-Camacho, "Evolution and ecology of locomotion in amphibians," in *Evolutionary Ecology of Amphibians.* CRC Press, 2023, pp. 135–157.

[47] C. Both and T. Grant, "Biological invasions and the acoustic niche: the effect of bullfrog calls on the acoustic signals of white-banded tree frogs," *Biology Letters*, vol. 8, no. 5, pp. 714–716, 2012. [Online]. Available: http://rsbl.royalsocietypublishing.org/cgi/doi/10.1098/rsbl.2012.0412

[48] S. Neckel-Oliveira and C. Gascon, "Abundance, body size and movement patterns of a tropical treefrog in continuous and fragmented forests in the Brazilian Amazon," *Biological Conservation*, vol. 128, no. 3, pp. 308–315, 2006.

[49] H. C. Gerhardt, R. E. Daniel, S. A. Perrill, and S. Schramm, "Mating behaviour and male mating success in the green treefrog," *Animal Behaviour*, vol. 35, no. 5, pp. 1490–1503, 1987.

[50] B. T. Szabó, S. L. Denham, and I. Winkler, "Computational models of auditory scene analysis: A review," *Frontiers in neuroscience*, vol. 10, 2016.

[51] C. S. Thakur, R. M. Wang, S. Afshar, T. J. Hamilton, J. C. Tapson, S. A. Shamma, and A. van Schaik, "Sound stream segregation: a neuromorphic approach to solve the "cocktail party problem" in real-time," *Frontiers in neuroscience*, vol. 9, 2015.

[52] L. Krishnan, M. Elhilali, and S. Shamma, "Segregating complex sound sources through temporal coherence," *PLoS computational biology*, vol. 10, no. 12, p. e1003985, 2014.

[53] R. W. Mill, T. M. Hohm, A. Bendixen, I. Winkler, and S. L. Denham, "Modelling the emergence and dynamics of perceptual organisation in auditory streaming," *PLoS computational biology*, vol. 9, no. 3, 2013.

[54] D. Barniv and I. Nelken, "Auditory streaming as an online classification process with evidence accumulation," *PloS one*, vol. 10, no. 12, p. e0144788, 2015.

[55] J. Rankin, E. Sussman, and J. Rinzel, "Neuromechanistic model of auditory bistability," *PLoS computational biology*, vol. 11, no. 11, 2015.

[56] D. L. Wang, "The time dimension for scene analysis," *IEEE Transactions on Neural Networks*, 2005.

[57] D. P. W. Ellis and D. F. Rosenthal, *Mid-level representations for computational auditory scene analysis.* Perceptual Computing Section, Media Laboratory, Massachusetts Institute of Technology, 1995.

[58] I. Winkler, S. L. Denham, and I. Nelken, "Modeling the auditory scene: predictive regularity representations and perceptual objects," *Trends in cognitive sciences*, vol. 13, no. 12, pp. 532–540, 2009.

[59] R. F. Lyon, "A computational model of filtering, detection and compression in the cochlea," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Paris, 1982, pp. 1282–1285.

[60] L. Wiegrebe and R. Meddis, "The representation of periodic sounds in simulated sustained chopper units of the ventral cochlear nucleus," *Journal of the Acoustical Society of America*, vol. 115, no. 3, pp. 1207–1218, 2004.

[61] R. D. Patterson, J. Holdsworth, and M. Allerhand, "Auditory models as preprocessors for speech recognition," *The Auditory Processing of Speech: from Auditory Periphery to Words*, pp. 67–89, 1992.

[62] R. Meddis, M. J. Hewitt, and T. M. Shackleton, "Implementation details of a computational model of the inner hair-cell/auditory-nerve synapse," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1813–1816, 1990.

[63] C. J. Sumner, E. A. Lopez-Poveda, L. P. O'Mard, and R. Meddis, "A revised model of the inner-hair cell and auditory-nerve complex," *The Journal of the Acoustical Society of America*, vol. 111, no. 5, pp. 2178–2188, 2002.

[64] R. O. Duda, R. F. Lyon, and M. Slaney, "Correlograms and the separation of sounds," in *Signals, Systems and Computers, 1990 Conference Record Twenty-Fourth Asilomar Conference on*, vol. 1. IEEE, 1990, p. 457.

[65] N. Madhu, A. Spriet, S. Jansen, R. Koning, and J. Wouters, "The potential for speech intelligibility improvement using the ideal binary mask and the ideal wiener filter in single channel noise reduction systems: Application to auditory prostheses," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 63–72, 2013.

[66] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan, "An auditory-based feature for robust speech recognition," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on.* IEEE, 2009, pp. 4625–4628.

[67] Y. Xu, J. B. Weaver, D. M. Healy, and J. Lu, "Wavelet transform domain filters: a spatially selective noise filtration technique," *IEEE transactions on image processing*, vol. 3, no. 6, pp. 747–758, 1994.

[68] Y. Li and D. Wang, "On the optimality of ideal binary time–frequency masks," *Speech Communication*, vol. 51, no. 3, pp. 230–239, 2009.

[69] A. Bendixen, I. SanMiguel, and E. Schröger, "Early electrophysiological indicators for predictive processing in audition: a review," *International Journal of Psychophysiology*, vol. 83, no. 2, pp. 120–131, 2012.

[70] S. L. Denham and I. Winkler, "The role of predictive models in the formation of auditory streams," *Journal of Physiology-Paris*, vol. 100, no. 1, pp. 154–170, 2006.

[71] J. Nix and V. Hohmann, "Combined estimation of spectral envelopes and sound source direction of concurrent voices by multidimensional statistical filtering," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 3, pp. 995–1008, 2007.

[72] D. Wang and P. Chang, "An oscillatory correlation model of auditory streaming," *Cognitive neurodynamics*, vol. 2, no. 1, pp. 7–19, 2008.

[73] S. A. Shamma, M. Elhilali, and C. Micheyl, "Temporal coherence and attention in auditory scene analysis," *Trends in neurosciences*, vol. 34, no. 3, pp. 114–123, 2011.

[74] M. Elhilali and S. A. Shamma, "A cocktail party with a cortical twist: how cortical mechanisms contribute to sound segregation," *The Journal of the Acoustical Society of America*, vol. 124, no. 6, pp. 3751–3771, 2008.

[75] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *The Journal of the Acoustical Society of America*, vol. 152, no. 1, pp. 107–151, 2022.

[76] B. Bahmei, E. Birmingham, and S. Arzanpour, "Cnn-rnn and data augmentation using deep convolutional generative adversarial network for environmental sound classification," *IEEE Signal Processing Letters*, vol. 29, pp. 682–686, 2022.

[77] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A survey of audio classification using deep learning," *IEEE Access*, 2023.

[78] H. A. Kassir, Z. D. Zaharis, P. I. Lazaridis, N. V. Kantartzis, T. V. Yioultsis, and T. D. Xenos, "A review of the state of the art and future challenges of deep learning-based beamforming," *IEEE Access*, vol. 10, pp. 80 869–80 882, 2022.

[79] N. Yalta, K. Nakadai, and T. Ogata, "Sound source localization using deep learning models," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 37–48, 2017.

[80] T. Ya, L. Yun, Z. Haoran, J. Zhang, W. Yu, G. Guan, and M. Shiwen, "Large-scale real-world radio signal recognition with deep learning," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 35–48, 2022.

[81] C. Hema and F. P. G. Marquez, "Emotional speech recognition using cnn and deep learning techniques," *Applied Acoustics*, vol. 211, p. 109492, 2023.

[82] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," *arXiv preprint arXiv:2006.00751*, 2020.

[83] B. Amjad, Q. Z. Ahmed, P. I. Lazaridis, M. Hafeez, F. A. Khan, and Z. D. Zaharis, "Radio slam: A review on radio-based simultaneous localization and mapping," *IEEE Access*, vol. 11, pp. 9260–9278, 2023.

[84] T. Zhang, H. Zhang, X. Li, J. Chen, T. L. Lam, and S. Vijayakumar, "Acousticfusion: Fusing sound source localization to visual slam in dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6868–6875.

[85] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *International conference on machine learning.* PMLR, 2019, pp. 4352–4362.

[86] D. Zhang, M. Ma, and L. Xia, "A comprehensive review on gans for time-series signals," *Neural Computing and Applications*, vol. 34, no. 5, pp. 3551–3571, 2022.

[87] R. Pichevar and J. Rouat, "Double-vowel segregation through temporal correlation: A bio-inspired neural network paradigm," in *ISCA Tutorial and Research Workshop on Non-Linear Speech Processing*, 2003.

[88] ——, "Monophonic sound source separation with an unsupervised network of spiking neurones," *Neurocomputing*, vol. 71, no. 1-3, pp. 109–120, 2007.

[89] F. Dramas, B. Oriola, B. G. Katz, S. J. Thorpe, and C. Jouffrais, "Designing an assistive device for the blind based on object localization and augmented auditory reality," in *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility.* ACM, 2008, pp. 263–264.

[90] R. K. Moore and B. Mitchinson, "A Biomimetic Vocalisation System for MiRo," pp. 1–12, 2017. [Online]. Available: http://arxiv.org/abs/1705.05472

[91] Y. Zouhir and K. Ouni, "A bio-inspired feature extraction for robust speech recognition," *SpringerPlus*, vol. 3, no. 1, p. 651, 2014.

[92] W. Yu, L. Jiajun, C. Ning, and Y. Wenhao, "Improved monaural speech segregation based on computational auditory scene analysis," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 2, 2013.

[93] S. Sundaram and S. Narayanan, "Discriminating two types of noise sources using cortical representation and dimension reduction technique," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I—-213.

[94] M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent, "Sound source localization," *European annals of otorhinolaryngology, head and neck diseases*, vol. 135, no. 4, pp. 259–264, 2018.

[95] Z. Wang, W. Zou, W. Zhang, H. Ma, C. Zhang, and Y. Guo, "Auditory feature driven model predictive control for sound source approaching," *International Journal of Control, Automation and Systems*, vol. 22, no. 2, pp. 676–689, 2024.

[96] A. Pico, G. Schillaci, V. V. Hafner, and B. Lara, "How do i sound like? forward models for robot ego-noise prediction," in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob).* IEEE, 2016, pp. 246–251.

[97] A. Schmidt, H. W. Löllmann, and W. Kellermann, "A novel ego-noise suppression algorithm for acoustic signal enhancement in autonomous systems," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 6583–6587.

[98] T. E. Tuncer and B. Friedlander, *Classical and modern direction-of-arrival estimation.* Academic Press, 2009.

[99] I. W. Selesnick and C. S. Burrus, "Generalized digital butterworth filter design," *IEEE Transactions on signal processing*, vol. 46, no. 6, pp. 1688–1694, 1998.

[100] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 6. IEEE, 2003, pp. VI–553.

[101] B. G. Gorshkov, K. YÃŒksel, A. A. Fotiadi, M. Wuilpart, D. A. Korobko, A. A. Zhirnov, K. V. Stepanov, A. T. Turov, Y. A. Konstantinov, and I. A. Lobach, "Scientific applications of distributed acoustic sensing: State-of-the-art review and perspective," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1033

[102] F. Sun, Q. Lu, S. Feng, and T. Zhang, "Flexible artificial sensory systems based on neuromorphic devices," *ACS Nano*, vol. 15, no. 3, pp. 3875–3899, 2021.

[103] F. Zhou and Y. Chai, "Near-sensor and in-sensor computing," *Nature Electronics*, vol. 3, no. 11, pp. 664–671, 2020.

[104] H. Taheri and Z. C. Xia, "Slam; definition and evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197620303092

[105] S. Michaud, S. Faucher, F. Grondin, J.-S. Lauzon, M. Labbe, D. Letourneau, F. Ferland, and F. Michaud, "3d localization of a sound source using mobile microphone arrays referenced by slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 402–10 407.

[106] W. Zhang, G. Han, X. Wang, M. Guizani, K. Fan, and L. Shu, "A node location algorithm based on node movement prediction in underwater acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3166–3178, 2020.

[107] C. Evers, H. W. Lollmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, and W. Kellermann, "The locata challenge: Acoustic source localization and tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1620–1643, 2020.

[108] M. Flottmann, M. Eisoldt, J. Gaal, M. Rothmann, M. Tassemeier, T. Wiemann, and M. Porrmann, "Energy-efficient fpga-accelerated lidar-based slam for embedded robotics," in *2021 International Conference on Field-Programmable Technology (ICFPT)*, 2021, pp. 1–6.

[109] A. K. Fuchs, C. Feldbauer, and M. Stark, "Monaural sound localization," in *INTERSPEECH*, 2011, pp. 2521–2524.

[110] H. Zhang, Y. Zhou, and L. Reiss, "Listening with two ears–new insights and perspectives in binaural research," *Frontiers in Neuroscience*, vol. 17, p. 1323330, 2023.

[111] B. A. Wright and M. B. Fitzgerald, "Different patterns of human discrimination learning for two interaural cues to sound-source location," *Proceedings of the National Academy of Sciences*, vol. 98, no. 21, pp. 12 307–12 312, 2001.

[112] J. Sodnik, R. Sušnik, M. Štular, and S. Tomažič, "Spatial sound resolution of an interpolated hrir library," *Applied Acoustics*, vol. 66, no. 11, pp. 1219–1234, 2005.

[113] F. Winter, H. Wierstorf, A. Raake, and S. Spors, "The two! ears database," in *Audio Engineering Society Convention 142*. Audio Engineering Society, 2017.

[114] M. Burkhard and R. Sachs, "Anthropometric manikin for acoustic research," *The Journal of the Acoustical Society of America*, vol. 58, no. 1, pp. 214–222, 1975.

[115] P. Majdak, Y. Iwaya, T. Carpentier, R. Nicol, M. Parmentier, A. Roginska, Y. Suzuki, K. Watanabe, H. Wierstorf, H. Ziegelwanger *et al.*, "Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions," in *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.

[116] H. Shukur, S. R. Zeebaree, A. J. Ahmed, R. R. Zebari, O. Ahmed, B. S. A. Tahir, and M. A. Sadeeq, "A state of art survey for concurrent computation and clustering of parallel computing for distributed systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 148–154, 2020.

[117] G. R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2002.

[118] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial Mathematics, 1987, vol. 5.

[119] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile Sensor Network deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02).* Citeseer, 2002, pp. 299–308.

[120] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein, "A Case Study in Building Layered DHT Applications," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 97–108.

[121] Q. Liang, X. Cheng, S. C. H. Huang, and D. Chen, "Opportunistic sensing in wireless sensor networks: theory and application," *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 2002–2010, 2014.

[122] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.

[123] Y. Demchenko, F. Turkmen, C. de Laat, C. Blanchet, and C. Loomis, "Cloud based big data infrastructure: Architectural components and automated provisioning," in *High Performance Computing & Simulation (HPCS), 2016 International Conference on.* IEEE, 2016, pp. 628–636.

[124] E. N. Alkhanak, S. P. Lee, and S. U. R. Khan, "Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities," *Future Generation Computer Systems*, vol. 50, pp. 3–21, 2015.

[125] B. Biskupski, J. Dowling, and J. Sacha, "Properties and Mechanisms of Self-Organizing MANET and P2P Systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 2, no. 1, p. 1, 2007.

[126] P. Horn, "Autonomic computing: IBM's Perspective on the State of Information Technology," 2001.

[127] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[128] G. Coulouris, J. Dollimore, and T. Kindberg, "Distributed Systems: Concepts and Design Edition 3," vol. 15, p. 4, 2001.

[129] T. Hogg and B. A. Huberman, "Controlling Chaos in Distributed Systems," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 6, pp. 1325–1332, 1991.

[130] I. Satoh, "MobileSpaces: A Framework for Building Adaptive Distributed Applications Using a Hierarchical Mobile Agent System," in *Distributed Computing Systems, 2000. Proceedings. 20th International Conference on.* IEEE, 2000, pp. 161–168.

[131] R. W. Brennan, M. Fletcher, and D. H. Norrie, "An Agent-Based Approach to Reconfiguration of Real-Time Distributed Control Systems," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 4, pp. 444–451, 2002.

[132] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, "Agent-based Design Model of Adaptive Distributed Systems," *Applied Intelligence*, vol. 9, no. 1, pp. 57–70, 1998.

[133] G. Tesauro, D. M. Chess, W. E. Walsh, R. Das, A. Segal, I. Whalley, J. O. Kephart, and S. R. White, "A Multi-Agent Systems Approach to Autonomic Computing," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 2004, pp. 464–471.

[134] A. J. Chakravarti, G. Baumgartner, and M. Lauria, "The Organic Grid: Self-Organizing Computation on a Peer-to-Peer Network," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 3, pp. 373–384, 2005.

[135] D. Kurzyniec, T. Wrzosek, D. Drzewiecki, and V. Sunderam, "Towards Self-Organizing Distributed Computing Frameworks: The H2O Approach," *Parallel Processing Letters*, vol. 13, no. 02, pp. 273–290, 2003.

[136] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl, "A Journey to Highly Dynamic, Self-Adaptive Service-Based Applications," *Automated Software Engineering*, vol. 15, no. 3-4, pp. 313–341, 2008.

[137] I. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. Dobre, A. Muraru, A. Costan, M. Dediu, and C. Stratan, "MonALISA: An Agent Based, Dynamic Service System to Monitor, Control and Optimize Distributed Systems," *Computer Physics Communications*, vol. 180, no. 12, pp. 2472–2498, 2009.

[138] S. R. White, J. E. Hanson, I. Whalley, D. M. Chess, and J. O. Kephart, "An Architectural Approach to Autonomic Computing," in *Autonomic Computing, International Conference on*. IEEE Computer Society, 2004, pp. 2–9.

[139] T. Máhr, J. Srour, M. de Weerdt, and R. Zuidwijk, "Can Agents Measure Up? A Comparative Study of an Agent-Based and On-Line Optimization Approach for a Drayage Problem with Uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 99–119, 2010.

[140] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Reading, MA, 1999, vol. 33.

[141] W. van der Hoek and M. Wooldridge, "Multi-Agent Systems," *Foundations of Artificial Intelligence*, vol. 3, pp. 887–928, 2008.

[142] M. Wooldridge, *An Introduction to Multiagent Systems.* Wiley, 2009.

[143] L. Panait and S. Luke, "Cooperative Multi-agent Learning: The State of the Art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[144] N. R. Jennings, "On Agent-based Software Engineering," *Artificial Intelligence*, vol. 117, no. 2, pp. 277–296, 2000.

[145] G. Eleftherakis, O. Paunovski, K. Rousis, and A. Cowling, "Emergent distributed bio-organization: A framework for achieving emergent properties in unstructured distributed systems," in *Intelligent Distributed Computing VI.* Springer, 2012, pp. 23–28.

[146] H. Schmeck, "Organic Computing - A New Vision for Distributed Embedded Systems," in *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on.* IEEE, 2005, pp. 201–203.

[147] E. Baccour, N. Mhaisen, A. A. Abdellatif, A. Erbad, A. Mohamed, M. Hamdi, and M. Guizani, "Pervasive ai for iot applications: A survey on resource-efficient distributed artificial intelligence," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2366–2418, 2022.

[148] N. Vlassis, "Multiagent systems and distributed ai," *Intelligent Autonomous Systems, Informatics Institute, University of Amsterdam*, 2003.

[149] N. Mungoli, "Scalable, distributed ai frameworks: Leveraging cloud computing for enhanced deep learning performance and efficiency," *arXiv preprint arXiv:2304.13738*, 2023.

[150] M. C. Huebscher and J. A. McCann, "A Survey of Autonomic Computing - Degrees, Models, and Applications," *ACM Computing Surveys (CSUR)*, vol. 40, no. 3, p. 7, 2008.

[151] O. Paunovski, G. Eleftherakis, and A. J. Cowling, "Disciplined Exploration of Emergence Using Multi-Agent Simulation Framework," *Computing and Informatics*, vol. 28, no. 3, pp. 369–391, 2012.

[152] R. W. Collier, E. O'Neill, D. Lillis, and G. M. O'Hare, "MAMS: Multi-Agent MicroServices," in *Companion Proceedings of the 2019 World Wide Web Conference.* ACM, 2019, p. 8. [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3308560.3316509

[153] C. Ihejimba and R. Z. Wenkstern, "A Cloud-Based Microservices Solution for Multi-Agent Traffic Control Systems," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024).* IFAAMAS, 2024, pp. 889–897. [Online]. Available: https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p889.pdf

[154] M. Jagutis, S. Russell, and R. Collier, "Using Multi-Agent MicroServices (MAMS) for Agent Based Modelling," *arXiv preprint arXiv:2307.14745*, 2023. [Online]. Available: https://arxiv.org/abs/2307.14745

[155] C. J. Amaral, S. P. Bernardes, M. Conceição, J. F. Hübner, L. P. A. Lampert, O. A. Matoso, and M. R. Zatelli, "Finding new routes for integrating Multi-Agent Systems using Apache Camel," *arXiv preprint arXiv:1905.10490*, 2019. [Online]. Available: https://arxiv.org/abs/1905.10490

[156] M. Lombardi, F. Pascale, and D. Santaniello, "Internet of things: A general overview between architectures, protocols and applications," *Information*, vol. 12, no. 2, p. 87, 2021.

[157] A. R. Naik and B. N. Keshavamurthy, "Next level peer-to-peer overlay networks under high churns: a survey," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 905–931, 2020.

[158] A. I. Ameur, A. Lakas, M. B. Yagoubi, and O. S. Oubbati, "Peer-to-peer overlay techniques for vehicular ad hoc networks: Survey and challenges," *Vehicular Communications*, vol. 34, 2022.

[159] A. Razzaq and S. A. Ghayyur, "A systematic mapping study: The new age of software architecture from monolithic to microservice architecture - awareness and challenges," *Computer Applications in Engineering Education*, vol. 31, no. 2, pp. 421–451, 2023.

[160] H. Siddiqui, F. Khendek, and M. Toeroe, "Microservices based architectures for iot systems-state-of-the-art review," *Internet of Things*, p. 100854, 2023.

[161] G. M. Hall, *Adaptive code: Agile coding with design patterns and SOLID principles.* Microsoft press, 2017.

[162] J. Thönes, "Microservices," *IEEE software*, vol. 32, no. 1, pp. 116–116, 2015.

[163] L. Baresi and M. Garriga, "Microservices: The evolution and extinction of web services?" *Microservices: Science and Engineering*, pp. 3–28, 2020.

[164] A. Lock, *ASP. NET core in Action.* Simon and Schuster, 2023.

[165] F. Henderson, "Software engineering at google," *arXiv preprint arXiv:1702.01715*, 2017.

[166] M. Śliwa and B. Pańczyk, "Performance comparison of programming interfaces on the example of rest api, graphql and grpc," *Journal of Computer Sciences Institute*, vol. 21, pp. 356–361, 2021.

[167] R. C. Cardoso and A. Ferrando, "A review of agent-based programming for multi-agent systems," *Computers*, vol. 10, no. 2, p. 16, 2021.

[168] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[169] C. Castelfranchi, F. Dignum, C. M. Jonker, and J. Treur, "Deliberative normative agents: Principles and architecture," in *Intelligent Agents VI. Agent Theories, Architectures, and Languages: 6th International Workshop, ATAL 99, Orlando, Florida, USA, July 15-17, 1999. Proceedings 6.* Springer, 2000, pp. 364–378.

[170] J. P. Müller, *The design of intelligent agents: a layered approach.* Springer Science & Business Media, 1996, vol. 1177.

[171] F. Meneguzzi and L. De Silva, "Planning in bdi agents: a survey of the integration of planning algorithms and agent reasoning," *The Knowledge Engineering Review*, vol. 30, no. 1, pp. 1–44, 2015.

[172] L. Braubach and A. Pokahr, "Developing distributed systems with active components and Jadex," *Scalable Computing: Practice and Experience*, vol. 13, no. 2, 2012.

[173] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[174] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[175] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A systematic review on supervised and unsupervised machine learning algorithms for data science," *Supervised and unsupervised learning for data science*, pp. 3–21, 2020.

[176] G. Mitrentsis and H. Lens, "Data-driven dynamic models of active distribution networks using unsupervised learning techniques on field measurements," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 2952–2965, 2021.

[177] D. J. Joshi, I. Kale, S. Gandewar, O. Korate, D. Patwari, and S. Patil, "Reinforcement learning: a survey," in *Machine Learning and Information Processing: Proceedings of ICMLIP 2020.* Springer, 2021, pp. 297–308.

[178] D. E. Kirk, *Optimal control theory: an introduction.* Courier Corporation, 2004.

[179] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[180] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[181] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, no. 2, 2000, p. 2.

[182] H. Zhang and T. Yu, "Taxonomy of reinforcement learning algorithms," *Deep reinforcement learning: Fundamentals, research and applications*, pp. 125–133, 2020.

[183] A. W. Beggs, "On the convergence of reinforcement learning," *Journal of economic theory*, vol. 122, no. 1, pp. 1–36, 2005.

[184] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[185] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.

[186] T. Eimer, M. Lindauer, and R. Raileanu, "Hyperparameters in reinforcement learning and how to tune them," in *International conference on machine learning*. PMLR, 2023, pp. 9104–9149.

[187] J. H. Holland, *Emergence: From chaos to order.* OUP Oxford, 2000.

[188] S. Stepney, F. A. C. Polack, and H. R. Turner, "Engineering emergence," in *Engineering of Complex Computer Systems, 2006. ICECCS 2006. 11th IEEE International Conference on.* IEEE, 2006, pp. 9—-pp.

[189] H. Sayama, *Introduction to the modeling and analysis of complex systems.* Open SUNY Textbooks, 2015.

[190] M. A. Bedau, "Weak emergence," *No{û}s*, vol. 31, no. s11, pp. 375–399, 1997.

[191] R. B. Laughlin, *A different universe: Reinventing physics from the bottom down.* Basic Books, 2006.

[192] J. I. N. S.-Y. H. Hong-Bing and F. A. N. Gao-Jun, "Emergence-Oriented Research on Multi-Agent Systems and Its State of Arts [J]," *Chinese Journal of Computers*, vol. 6, p. 1, 2008.

[193] R. Abbott, "Emergence explained: Abstractions: Getting epiphenomena to do real work," *Complexity*, vol. 12, no. 1, pp. 13–26, 2006.

[194] D. Foster, *Generative deep learning.* " O'Reilly Media, Inc.", 2022.

[195] J. Cheng, Y. Yang, X. Tang, N. Xiong, Y. Zhang, and F. Lei, "Generative adversarial networks: A literature review," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 12, pp. 4625–4647, 2020.

[196] D. Pappas, "Implementing Emergent Distributed Bio-Organization," University of Sheffield, Tech. Rep., 2013.

[197] ——, "Emergent Distributed Bio Organization: A Middleware for the Internet of Things," Master's thesis, 2014.

[198] A. R. Pollastri, J. S. Ablon, and M. J. Hone, *Collaborative Problem Solving*. Springer, 2019.

[199] L. M. Nelson, "Collaborative problem solving," in *Instructional-design theories and models*. Routledge, 2013, pp. 241–267.

[200] A. C. Graesser, S. M. Fiore, S. Greiff, J. Andrews-Todd, P. W. Foltz, and F. W. Hesse, "Advancing the science of collaborative problem solving," *Psychological Science in the Public Interest*, vol. 19, no. 2, pp. 59–92, 2018.

[201] J. Allen, C. M. Teng, and L. Galescu, "Dialogue as collaborative problem solving: A case study," *Advances in Cognitive Systems*, vol. 7, no. 2019, pp. 195–214, 2018.

[202] N. Blaylock, J. Allen, and G. Ferguson, "Managing communicative intentions with collaborative problem solving," *Current and new directions in discourse and dialogue*, pp. 63–84, 2003.

[203] N. M. Dowell, Y. Lin, A. Godfrey, and C. Brooks, "Exploring the relationship between emergent sociocognitive roles, collaborative problem-solving skills, and outcomes: A group communication analysis." *Journal of Learning Analytics*, vol. 7, no. 1, pp. 38–57, 2020.

[204] A. C. Graesser, S. Greiff, M. Stadler, and K. T. Shubeck, "Collaboration in the 21st century: The theory, assessment, and teaching of collaborative problem solving," p. 106134, 2020.

[205] C. Sun, V. J. Shute, A. Stewart, J. Yonehiro, N. Duran, and S. D'Mello, "Towards a generalized competency model of collaborative problem solving," *Computers & Education*, vol. 143, p. 103672, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131519302258

[206] M. Zhang, Q. Bai, F. Ren, and J. Fulcher, "Collaborative agents for complex problems solving," *Computational Intelligence: Collaboration, Fusion and Emergence*, pp. 361–399, 2009.

[207] W. Pedrycz and P. Rai, "A multifaceted perspective at data analysis: A study in collaborative intelligent agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 4, pp. 834–844, 2009.

[208] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *arXiv preprint arXiv:2309.07864*, 2023.

[209] E. H. Durfee, "Cooperative distributed problem solving between (and within) intelligent agents," in *Neuroscience: From Neural Networks to Artificial Intelligence: Proceedings of a US-Mexico Seminar held in the city of Xalapa in the state of Veracruz on December 9–11, 1991.* Springer, 1993, pp. 84–98.

[210] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.

[211] J. Andrews-Todd and C. M. Forsyth, "Exploring social and cognitive dimensions of collaborative problem solving in an open online simulation-based task," *Computers in Human Behavior*, vol. 104, p. 105759, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563218305156

[212] K. Y. Yin, A. G. K. Abdullah, and N. J. Alazidiyeen, "Collaborative problem solving methods towards critical thinking." *International Education Studies*, vol. 4, no. 2, pp. 58–62, 2011.

[213] T. J. Nokes-Malach, M. L. Meade, and D. G. Morrow, "The effect of expertise on collaborative problem solving," *Thinking & Reasoning*, vol. 18, no. 1, pp. 32–58, 2012.

[214] I. A. Novikova, "Big five (the five-factor model and the five-factor theory)," *The encyclopedia of cross-cultural psychology*, vol. 1, no. 3, 2013.

[215] F. Durupinar, N. Pelechano, J. Allbeck, U. GÃŒdÃŒkbay, and N. I. Badler, "How the ocean personality model affects the perception of crowds," *IEEE Computer Graphics and Applications*, vol. 31, no. 3, pp. 22–31, 2011.

[216] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating heterogeneous crowd behaviors using personality trait theory," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*, 2011, pp. 43–52.

[217] B. L. Putro, Y. Rosmansyah, and S. Suhardi, "An intelligent agent model for learning group development in the digital learning environment: A systematic literature review," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1159–1166, 2020.

[218] Á. Carrera and C. A. Iglesias, "A systematic review of argumentation techniques for multi-agent systems research," *Artificial Intelligence Review*, vol. 44, pp. 509–535, 2015.

[219] X. Fan and J. Yen, "Modeling and simulating human teamwork behaviors using intelligent agents," *Physics of life reviews*, vol. 1, no. 3, pp. 173–201, 2004.

[220] Y. Wang and V. Chiew, "On the cognitive process of human problem solving," *Cognitive systems research*, vol. 11, no. 1, pp. 81–92, 2010.

[221] K. Hoffman and C. Elwin, "The relationship between critical thinking and confidence in decision making," *Australian Journal of Advanced Nursing, The*, vol. 22, no. 1, pp. 8–12, 2004.

[222] M. Fisher and F. C. Keil, "The curse of expertise: When more knowledge leads to miscalibrated explanatory insight," *Cognitive science*, vol. 40, no. 5, pp. 1251–1269, 2016.

[223] O. Askarisichani, E. Y. Huang, K. S. Sato, N. E. Friedkin, F. Bullo, and A. K. Singh, "Expertise and confidence explain how social influence evolves along intellective tasks," *arXiv preprint arXiv:2011.07168*, 2020.

[224] H. Wierstorf, M. Geier, A. Raake, and S. Spors, "A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances," Jun. 2016. [Online]. Available: https://doi.org/10.5281/zenodo.55418

[225] "Rt60 calculator wallace c. sabine calculation reverb time reverberation time sabin sanine's formula online sound pressure sound level - sengpielaudio sengpiel berlin," https://sengpielaudio.com/calculator-RT60.htm, accessed: 2024-04-27.

[226] "Iceberg - sheffield hpc documentation," https://docs.hpc.shef.ac.uk/en/latest/decommissioned/iceberg.html, accessed: 2024-04-27.

[227] H. Pontynen and N. H. Salminen, "Resolving front-back ambiguity with head rotation: The role of level dynamics," *Hearing Research*, vol. 377, pp. 196–207, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378595518305550

[228] S. T. Birchfield and R. Gangishetty, "Acoustic localization by interaural level difference," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 4. IEEE, 2005, pp. iv–1109.

[229] J. E. Schneider, "Energy balance and reproduction," *Physiology & behavior*, vol. 81, no. 2, pp. 289–317, 2004.

[230] R. Pascal and A. Pross, "Stability and its manifestation in the chemical and biological worlds," *Chemical Communications*, vol. 51, no. 90, pp. 16 160–16 165, 2015.

[231] W. E. Roberts, *Evolution and ecology of arboreal egg-laying frogs.* University of California, Berkeley, 1994.

[232] J. Vasarhelyi, O. M. Salih, H. M. Rostum, and R. Benotsname, "An overview of energies problems in robotic systems," *Energies*, vol. 16, no. 24, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/24/8060

[233] D. Hrubý, D. Marko, M. Olejár, V. Cviklovič, and D. Horňák, "Comparison of control algorithms by simulating power consumption of differential drive mobile robot motion control in vineyard row," *Acta Technologica Agriculturae*, vol. 24, no. 4, pp. 195–201, 2021.

[234] A. Liu, H. Liu, B. Yao, W. Xu, and M. Yang, "Energy consumption modeling of industrial robot based on simulated power data and parameter identification," *Advances in Mechanical Engineering*, vol. 10, no. 5, p. 1687814018773852, 2018.

[235] J. S. Ulloa, T. Aubin, D. Llusia, É. A. Courtois, A. Fouquet, P. Gaucher, S. Pavoine, and J. Sueur, "Explosive breeding in tropical anurans: environmental triggers, community composition and acoustic structure," *BMC ecology*, vol. 19, pp. 1–17, 2019.

[236] N. D. Theise and M. C. Kafatos, "Complementarity in biological systems: a complexity view," *Complexity*, vol. 18, no. 6, pp. 11–20, 2013.

[237] S. Gu and G. Mao, "An improved q-learning algorithm for path planning in maze environments," in *Intelligent Systems and Applications*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2021, pp. 547–557.

[238] S. Thrun and A. Schwartz, "Issues in using function approximation for reinforcement learning," in *Proceedings of the 1993 connectionist models summer school.* Psychology Press, 2014, pp. 255–263.

[239] B. Xu, Q. Zhou, J. Shi, and S. Li, "Hierarchical q-learning network for online simultaneous optimization of energy efficiency and battery life of the battery/ultracapacitor electric vehicle," *Journal of Energy Storage*, vol. 46, p. 103925, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X21015899

[240] E. Even-Dar, Y. Mansour, and P. Bartlett, "Learning rates for q-learning." *Journal of machine learning Research*, vol. 5, no. 1, 2003.

[241] V. Kosana, K. Teeparthi, S. Madasthu, and S. Kumar, "A novel reinforced online model selection using q-learning technique for wind speed prediction," *Sustainable Energy Technologies and Assessments*, vol. 49, p. 101780, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138821007943

[242] B. Stout, “Smart moves: Intelligent pathfinding,” *Game developer magazine*, vol. 10, pp. 28–35, 1996.

[243] Y. Sakurai, T. Miyake, and K. Hiraishi, “Interactive learning of social agents based on confidence degree,” in *2012 IEEE International Conference on Fuzzy Systems*. IEEE, 2012, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/6251234/

[244] J. Sabater, M. Paolucci, and R. Conte, “A fuzzy system based approach to social modeling in multi-agent systems,” in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2002, pp. 524–525. [Online]. Available: https://dl.acm.org/doi/10.1145/544741.544850

[245] R. Falcone and C. Castelfranchi, “Devising a trust model for multi-agent interactions using confidence and reputation,” *Applied Artificial Intelligence*, vol. 18, no. 9-10, pp. 769–783, 2004. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/08839510490509045

[246] M. Cossentino, L. Sabatucci, and A. Chella, “Modeling social aspects of multi-agent systems: The aml approach,” in *Agent-Oriented Software Engineering VI*. Springer, 2006, pp. 57–69.

Appendix A

Simulation and Experiment Parameters

Treefrogs - synthetic call parameters

The following table describes the values that the treefrog synthetic call function has been tested with and found empirically to be true to real-world treefrog mating call samples. High and low values for experimentation are provided, as well as the numbers used for the two different target types that have been used in the experiments: those with low frequency, and those with high frequency. These two targets were used due to the ease of extracting base frequency and setting them as preferences later for the trackers.

Parameter	Low	High	Low freq. target	High freq. target
Attack time	5	50	40	
Decay time	25	75	40	
Gap time	20			
Start pitch	500	1300	700	950
End pitch	500	1300	1050	1300
Harmonics	$[0.3, 1.0, 0.7, 0.3, 0.3]$			
Harmonics	$0.1 + 0.1$	$0.1 + 1.0$	1	1
Number of pulses	5	15	10	10

Acoustic scene - binaural simulator parameters

This table provides the values used for the binaural simulator from Two!Ears that describe the acoustic scene for the purposes of generating impulse responses. The parameter names are exactly the ones used in the toolkit for defining the non-default values. These values have been used across all experiments to not affect strategy performance otherwise. From the child objects of the `simulator` given values, `Sinks` refers to the listener, while the `rotateAroundZ` function is used to define the orientation of the listener. Values named **scenario_*** refer to values taken from the experiment, specifically listener and speaker location in **X** and **Y** and orientation of the entity in the room with **O**.

Object	Parameters	Values
`simulator`	`MaximumDelay`	0.05
	`PreDelay`	0
`simulator.Sources`	`Name`	Target
	`Position`	[scenario_X, scenario_Y, 0.2]
`simulator.Sinks`	`Name`	Tracker
	`Position`	[scenario_X, scenario_Y, 0.2]
`rotateAroundZ`	`Sources`	scenario_O
	`Sinks`	scenario_O
`simulator.Room`	`Name`	Room
	`Position`	$[0, 0, 0]$
	`UnitX`	$[1, 0, 0]$
	`UnitY`	$[0, 0, 1]$
	`LengthX`	10.0
	`LengthY`	10.0
	`LengthZ`	3.0
	`ReverberationMode`	2D
	`ReverberationMaxOrder`	8
	`RT60`	1.0
	`ReflectionCoeffs`	$[0.1, 0.1, 0.1, 0.1, 0.1]$
	`AbsorbtionCoeffs`	$[0, 0, 0, 0, 0]$

Obstacles - recursive division parameters

This defines the number of iterations required by the recursive division to provide the desired outcome. Iterations include a step of "divisions" of the walls and then "opening doors" to ensure there is access everywhere. All experiments use one of these two values, or both in the case of highly dynamic experiments.

Density	Iterations	Divisions	Openings
few	1	1	3
many	3	3	9

Highly dynamic environment - simulation parameters

The highly dynamic environment experiments have been defined as those where the obstacles increase over time, and the number of targets also change over time. The following table shows at which time intervals in seconds during simulation runtime the changes occur. The time intervals randomiser selects a number from the bounds presented below. Low vs high interest for the targets means that the target has a base frequency farther than the preferred one of the trackers which is assigned at creation runtime. These values are governed by the high/low frequency target settings presented earlier in this Appendix.

Time from start	**Action**
$15s - 25s$	2nd target appears (of low interest)
$25s - 50s$	2nd target leaves
$40s - 60s$	Obstacle density increase
$50s - 75s$	3rd target leaves (of high interest)

Socio-cognitive traits - value ranges

The following table shows the value ranged for the different high/low cases used in simulations for the socio-cognitive traits. For the social ones (inquisitive, sharing) these are the probabilities to take the respective action during agent reasoning in the new decision-making layer introduced. These value ranges were chosen to have significant divergence from each other and thereby get meaningful results.

Trait	**Value**	**Value range**
`ProblemSolving`	*high*	Adaptive strategy
	low	Combined strategy
`SelfConfidence`	*high*	$0.6 - 0.9$
	low	$0.1 - 0.4$
`Inquisitive`	*high*	$0.6 - 0.9$
	low	$0.1 - 0.4$
`Sharing`	*high*	$0.6 - 0.9$
	low	$0.1 - 0.4$