POLITECNICO DI TORINO

POLITECNICO DI TORINO

Master Degree course in Mechatronic Engineering

Master Degree Thesis

Brain Emotional Sliding Mode Control for a Heterogeneous Multi-agent System.

Supervisors
Prof. Elisa Capello
Prof. Jonatan Uziel Alvarez Muñoz
Prof. Juan Antonio Escareno

Candidate
Francesco Fracasso

Academic Year 2022-2023

Acknowledgements

To begin, I would like to thank my family for their support and patience throughout these years of the university. I owe my greatest achievements to them.

A heartfelt thank you to my friends with whom I shared this journey and who helped me alleviate the hardships and amplify the joys. Thank you for always being by my side.

An immense thank you also goes to my new friends, the Pastis Boys, who entered my heart in just a few months and will never leave. Thank you for accompanying me during a complicated and tense period and making it light and carefree.

Now, I would like to express my gratitude to those who actively contributed to the completion of this thesis. A special thanks to Professor Capello for her availability, and patience, and for sharing her profound knowledge with me.

A warm thank you to all the people in LAB-015 at IPSA for welcoming me and making me feel part of a group, and to professors Alvarez and Escareno for supervising and guiding me throughout this experience.

Abstract

The ability to adapt to a continuously changing environment and robustness to disturbances and uncertainties are always crucial features for a controller. In the last decades, an increasing interest in modeling the Limbic system in the human brain led to the development of the Brain Emotional Learning-Based Intelligent Controller (BELBIC) in order to cope with unknown dynamics.

The novelty brought by this system, compared to a Neural Adaptive, is the ability to have a memory of past experiences, learn from them and adapt itself, providing a faster response when the same scenario is proposed again. In this thesis, the BELBIC controller is used to control a heterogeneous Multi-Agent System composed of a fleet of drones following a mobile robot leader, both for consensus and formation purposes.

At first, the biological review of the Limbic system, which inspires the BELBIC, is described to understand its behavior and thus its computational model developed in the literature. This model is obviously simpler than the real Limbic system yet contains its main crucial parts as *Thalamus, Sensory Cortex, Orbitofrontal Cortex, and Amygdala.*

The role of the sensory signal is to provide information to the Amygdala and Orbitofrontal Cortex and this is done by feeding these areas with knowledge about the tracking error. This information is carried by the *sliding surface* and hence, a theory review is provided in order to understand its ability to deal with uncertainties. Throughout this thesis, advanced types of sliding surfaces are considered and analyzed.

Finally, the results of the heterogeneous Multi-Agent System are presented, showing that the BELBIC provides better performances compared to other types of neural adaptive control.

Contents

List of Figures 4

List of Tables 6

1 Introduction 7
- 1.1 Sliding Mode Controller . . . 9
  - 1.1.1 Theory of SMC . . . 9
  - 1.1.2 Chattering phenomenum . . . 12
  - 1.1.3 Example . . . 12
- 1.2 Multi-Agent System . . . 14
  - 1.2.1 Graph Theory . . . 16
  - 1.2.2 Example . . . 18
- 1.3 Neural Network . . . 19
  - 1.3.1 Computational model . . . 19
  - 1.3.2 Radial Basis Function . . . 21
  - 1.3.3 Multi-layer with error back-propagation . . . 22
  - 1.3.4 Considerations and Drawbacks of the back-propagation . . . 24

2 LIMBIC System 25
- 2.1 Biological Model . . . 25
  - 2.1.1 Components of the Limbic System . . . 25
- 2.2 Computational Model . . . 26
  - 2.2.1 Amygdala . . . 28
  - 2.2.2 Orbitofrontal Cortex . . . 28
  - 2.2.3 Results of the *Acquisition* process . . . 29
- 2.3 Reward and Sensory Signal . . . 30

3 Neural Adaptive SMC vs NFTSMC 35
- 3.1 Terminal Sliding Mode Control . . . 35
  - 3.1.1 Terminal attractor . . . 35
- 3.2 Fast Terminal Sliding Mode Control . . . 37
- 3.3 Non-Singular Terminal Sliding Mode Control . . . 38
- 3.4 Non-Singular Fast Terminal Sliding Mode Control . . . 38
- 3.5 Example . . . 38

4 Brain Emotional Based Controller 47
4.1 Introduction 47
4.2 Comparison BELBIC vs Neural Adaptive 51
4.3 Control of a mobile robot via BELBIC 54
4.4 Control of a drone via BELBIC 57
4.5 Attitude and Position control of a quadcopter 60
4.5.1 Position Control block via BELBIC 61
4.5.2 Attitude Control block via BELBIC 62
4.6 Control of Heterogeneous MAS via BELBIC 64

5 Conclusion 75

Bibliography 77

List of Figures

1.1 Phase-portrait in sliding mode control. 10
1.2 Chattering phenomena due to discontinuous term in the control law . . . 12
1.3 Pendulum response whether a disturbance is applied or not. 14
1.4 Pendulum response with $\rho = 1$,$L = 2$. 14
1.5 Pendulum response with disturbance and $sign(s)$. 15
1.6 Pendulum response with disturbance and $\tanh s$. 16
1.7 A Hierarchical Agent Architecture 16
1.8 Directed and undirected graph. 17
1.9 Example of *Hierarchical* Multi-Agent system. 18
1.10 Artificial Neural Network model. 20
1.11 Influence of c and σ on the *Gaussian* 22
1.12 Scheme of the *learning* process. 22
1.13 *Local* and *Global* minimum. 23

2.1 Limbic System. 27
2.2 Simplified model of the Limbic System: Thalamus, Sensory Cortex, Orbitofrontal Cortex, and Amygdala. 27
2.3 Results of the *Acquisition* process. From up to bottom: Thalamus, Sensory Signal, Output E, Reward, V_{th}, V, W. 30
2.4 Conditioned Response(CR) and Unconditioned Response. 31
2.5 Control scheme with BELBIC. 32
2.6 Simulink Implementation of the control scheme. 32
2.7 Simulink scheme for Amygdala and OFC. 33

3.1 Mass-Spring-Damper System. 39
3.2 Tracking performance of Classical Sliding Mode Control. 41
3.3 Tracking performance of Terminal Sliding Mode Control. 42
3.4 Tracking performance of Fast Terminal Sliding Mode Control. 43
3.5 Tracking performance of Non-Singular Terminal Sliding Mode Control. . . 44
3.6 Tracking performance of Non-Singular Fast Terminal Sliding Mode Control. 45

4.1 BELBIC scheme. 48
4.2 General control scheme with BELBIC. 49
4.3 Tracking performance with no disturbance, BELBIC (a) and Neural Adaptive (b). 53

4.4 Learning weights of Amygdala (a) and Orbitofrontal Cortex (b). 54
4.5 Tracking performance with disturbance, BELBIC (a) and Neural Adaptive (b). 55
4.6 Representation of mobile robot. 56
4.7 Control scheme of a mobile robot. 57
4.8 Control results of the mobile robot. 58
4.9 Weights of the BELBIC neurons: V,Amygdala (a) and W,OFC (b). 59
4.10 Rotation of a reference frame. 60
4.11 General Control scheme of a drone. 61
4.12 Attitude and position of the drone via BELBIC with NTSMC. 65
4.13 Weights of BELBIC neurons for the x position, Amygdala (a) and OFC (b). 66
4.14 Etherogeneous Multi-Agent System. 67
4.15 Simulink scheme of MAS. 68
4.16 Position of the mobile leader. 69
4.17 x position of the agents. 70
4.18 y position of the agents. 71
4.19 z position of the agents. 72
4.20 Formation of the Heterogeneous MAS. 73
4.21 Zoom of the MAS configuration. 74

List of Tables

3.1 Comparison of Control Techniques 40

4.1 Comparison of BELBIC and Neural Adaptive 52
4.2 Position errors and control signals in a drone. 64
4.3 Attitude errors and control signals in a drone. 64

Chapter 1

Introduction

"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change."

In 1963, prof. Leon C. Megginson used this sentence to describe his thoughts on "*On the Origin of Species*", the masterpiece written by Charles Darwin in 1958 and which gave a new perspective on the history of human evolution.

According to his theory, the nature that we live in nowadays is the result of an evolutionary process that started at the beginning of natural history and never stopped. All the species that we can find in nature have undergone this process and their survival is mainly due to their capability to adapt to a continuously changing environment.

Adapting to changes is not an easy task, it requires another important process, the learning process, which distinguishes an intelligent species from a weaker one. Through thousands of years, the human brain has evolved and adapted to several scenarios, becoming a very complex system, and in order to cope with the learning process, it has developed a wide memory able to store past experiences and learn from them.

The main area in the human brain related to these functions is called the Limbic system.

The Limbic system, commonly known as the "*emotional brain*", is a network of interconnected brain structures that plays a crucial role in the human central nervous system. This complex system is involved in multiple functions, including the regulation of emotions, behavioral responses, memory, and learning. Over the years, understanding the Limbic system and its impact on the human mind and behavior has been the subject of intense scientific research.

As described by Mowrer (Mowrer, 1973) [1], the learning process is implemented by the amygdala-orbitofrontal system and it is based on emotional cues. This system operates within a two-process model of learning. It first evaluates incoming emotionally charged stimuli and then utilizes this evaluation as a reinforcer for stimulus-response learning. One of the key advantages of this approach, as noted by Rolls [2], is the clean separation between the motivation to respond and the response itself.

Thus, emotions and rewards (reinforcers) play a crucial role in the learning process since they are strictly related to the *Amygdala*, mainly responsible for human memory.

In the field of autonomous systems engineering, the Limbic system has garnered attention as a source of inspiration for the development of advanced control algorithms. In particular, the BELBIC (Brain Emotional Learning-Based Intelligent Controller) has generated significant interest in controlling complex systems, such as a fleet of drones.

The BELBIC controller is based on an innovative approach that seeks to emulate the functioning of the human Limbic system. This controller integrates emotional and cognitive signals to guide and learn the behavior of the controlled system. A computational two-process model of the Limbic system has been developed [3] and used to control SISO and MIMO systems.

The main objective of the engineer is to properly choose and design the inputs of the Limbic system: *Sensory Signal* and *Reward*. In [4], the BELBIC controller has been used with different dynamics as a robotic arm, submarine, gas turbine generator, and sensor fusion, demonstrating the heterogeneity of such a system.

The stability proof of the closed-loop system has been developed in [5] and [6], where the global asymptotically stability is based on the Lyapunov stability theory. In [7–9], the navigation and tracking problem have been successfully accomplished for several systems such as a mobile robot and robot arm manipulator. The BELBIC controller has also been used for flocking, consensus, and formation control for a Multi-Agent system, mainly applied to a swarm of drones, showing good results even when the number of agents is very high and the presence of obstacles is taken into consideration [10–12].

Categorized as *Reinforcement Learning*, the model of the LIMBIC system considering *reward* and *sensory signals* can have a complexity of $O(n*log^2(n))$ and in any case greater than $O(n)$, [13].

A single *neural network*(NN) hidden layer with *Radial Basis Functions* (RBF) as activation functions, is known to be a universal function estimator and so capable of reproducing most of the smooth functions [14], leading to *NeuralAdaptive* RBFNN control, which has been developed and used in several scenarios [14, 15]. A new perspective for the BELBIC is given by introducing NNRBF into the model, with a single hidden layer providing a complexity of $O(n)$.

In this new fashion, the BELBIC can be used to model the unknown or uncertain dynamic of the system to be controlled. Once the dynamic is learned, a classical control law can be applied in order to support the Limbic system.

The *neural network* is introduced to mimic the biological nature of the brain and the adaption laws are derived from Lyapunov functions for both *Amygdala* and *Orbitofrontal Cortex* since their nature is different compared to the *Neural Adaptive* one [16].

The Neural Adaptive can be seen as a simplified model of the BELBIC, for example considering only the Orbitofrontal Cortex in the model, thus providing worse results. A numerical example will be given in this thesis, comparing the BELBIC with Neural Adaptive, showing that the presence of the Amygdala is necessary to improve the results since it is mainly related to memory and experiences.

By utilizing concepts like sliding surfaces(used as sensory signals), neural networks, and adaption laws, the BELBIC controller can learn and adapt to different situations and specific system requirements without requiring complete knowledge of the system to be controlled. In this work, a *non-singular fast terminal sliding surface* (NFTSM)will be

considered as input of the neural network defining *Amygdala* and *Orbitofrontal Cortex*, that will estimate the unknown dynamic of a Multi-Agent System (MAS), in this case, drones.

1.1 Sliding Mode Controller

In an ideal world, simple states feedback would be enough to stabilize and control systems and there will not be discrepancies between simulations and real-world applications. Unknown or unmodelled dynamics, parameter uncertainty, and external disturbances lead to the necessity of robust control, i.e. a system able to reach the desired state while dealing with all the uncertainties.

The *Sliding Mode Controller*(SMC) has been developed to solve this problem and gained notoriety due to its ability to handle bounded uncertainties.

According to the stat-of-the-art review in [17], the study of SMC began in the late 1950s [18], but it was only after 1975 that its theory has been completed and clearly stated, thanks to [19, 20].

The first step is to define the so-called sliding surface, which is a subset of the state domain that ensures the system's motion meets the desired design specifications.

The second step involves selecting a control law that attracts the system state towards the sliding manifold, even in the presence of external and internal disturbances or uncertainties.

By following these two steps, the sliding mode controller ensures that the system operates in a stable and controlled manner. The switching function determines the behavior of the system on the sliding manifold, while the control law guides the system towards this manifold, compensating for disturbances and uncertainties.

This approach of combining a switching function and a control law allows the sliding mode controller to achieve robustness and stability in the face of uncertainties. It provides a powerful tool for controlling systems in various fields, such as robotics, aerospace, and power electronics, where disturbances and uncertainties are common challenges.

1.1.1 Theory of SMC

Supposing the system output $y(t)$ needs to track a desired reference signal $r(t)$; the tracking error is defined as $e = y(t) - r(t)$.

Definition 1 *The sliding surface is*

$$S(t) \doteq x \in \Re : \boldsymbol{s}(x,t) = 0$$

where the function $\boldsymbol{s}$*:* $\Re^{(n+1)} \to \Re$

$$\boldsymbol{s}(x,t) \doteq e^{\gamma-1} + k_\gamma e^{\gamma-2} + \cdots + k_2 e \tag{1.1}$$

where γ *is the relative degree between external and internal dynamics, with the latter one assumed to be asymptotically stable, and* k_i *are gains to be chosen.*

Consider an *affine*,γ-order SISO system

$$x^{(\gamma)} = f(x) + g(x)u + d(x,t) \tag{1.2}$$

Where x is a row vector defined as:

$$x = \begin{bmatrix} x & \dot{x} & \dots & x^{(\gamma-1)} \end{bmatrix}^\top$$

are the states of the system defined by smooth and derivable functions, u is the *control law* to be designed and $d(x,t)$ is a bounded external disturbance such that $|d(x,t)| \leq L > 0$.

The *control law* u, should be chosen in such a way that sliding surface is *invariant* and *attractive*, which means that it should drive the state toward the desired one and keep it on the surface once the tracking error is equal to 0, [21].

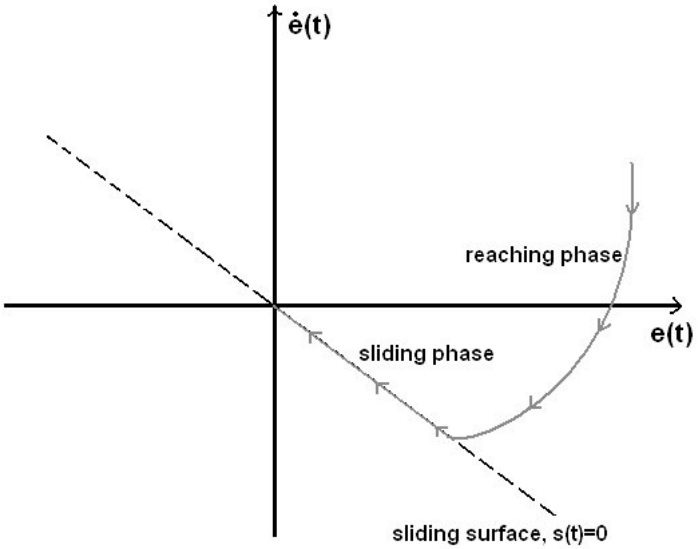


Figure 1.1: Phase-portrait in sliding mode control.

A Lyapunov-based approach can be used to design the control law. A candidate Lyapunov function providing asymptotically stability should satisfy two conditions:

1. Its derivative must be strictly negative
2. It must radially unbounded

One can choose the following equation as a function candidate, which satisfies the second condition

$$V = \frac{1}{2}s^2 \tag{1.3}$$

, and in order to obtain convergence to the desired states in a finite time, the first condition can be expressed as

$$\dot{V} \leq -\alpha V^{1/2} \quad , \alpha > 0 \tag{1.4}$$

Indeed, by integrating the equation, it is possible to obtain the reaching time

$$t_r \leq \frac{2V^{1/2}(0)}{\alpha} \tag{1.5}$$

The main step is to try to extract the control law from the Lyapunov function in order to guarantee a finite-time convergence toward the *invariant* and *attractive* sliding surface.

To guarantee that the sliding surface is invariant, the first term of the control law can be found by considering

$$\dot{s} = 0$$

$$\dot{s} = r^{(\gamma)} - f(x) - g(x)u + d(x,t) + k_\gamma e^{(\gamma-1)} + \cdots + k_2\dot{e} = 0 \tag{1.6}$$

$$u = \frac{1}{g(x)}\left(f(x) - r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2\dot{e}\right) \tag{1.7}$$

.

To guarantee the sliding surface is attractive and converges in a finite time, the eq. 1.4 must be satisfied, and so another term v must be added to the previous control law. The whole control law can be written as

$$u = \frac{1}{g(x)}(f(x) + u_{eq} + v) \tag{1.8}$$

, where

$$u_{eq} = -r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2\dot{e}$$

Considering the time derivative of eq.1.3

$$\dot{V} = s\dot{s} \tag{1.9}$$

By substituting eq.1.7, eq.1.8, eq.1.6 into eq.1.9, one can obtain

$$\dot{V} = s(d(x,t) + v) = sd(x,t) + sv \leq |s|L + sv \tag{1.10}$$

Every number can be written as

$$sign(x) = \frac{|x|}{x} = \frac{x}{|x|} \quad , x \neq 0$$

Thus, a good choice of v could be $v = -\rho sign(s)$, so that

$$\dot{V} =\leq |s|L - |s|\rho = -|s|(\rho - L) \tag{1.11}$$

Recalling the definition of the Lyapunov function in eq.1.3 and using it in 1.4, is it possible to obtain

$$\dot{V} =\leq -|s|(\rho - L) = -\frac{\alpha}{\sqrt{2}}|s| \tag{1.12}$$

Now, it is possible to define the second term of the control law

$$\rho = L + \frac{\alpha}{\sqrt{2}} \tag{1.13}$$

The *control law* in eq, 1.8 is defined such that the sliding surface has the required *invariant* and *attractive* properties. The first one is managed by u_{eq}, whereas the second one is by v, whose ρ parameters allow to cope with bounded uncertainties and finite reaching time.

1.1.2 Chattering phenomenum

The sliding mode control has several advantages as high control performance and robustness to model uncertainty and disturbances, but it requires the model to be affine in the control law. The main drawback, though, is the so-called *chattering* which arises from the discontinuous term $v = -\rho sign(s)$ and it manifests itself as high-frequency oscillations in both the sliding surface and the control input, [22], as shown by the following figure, [23]

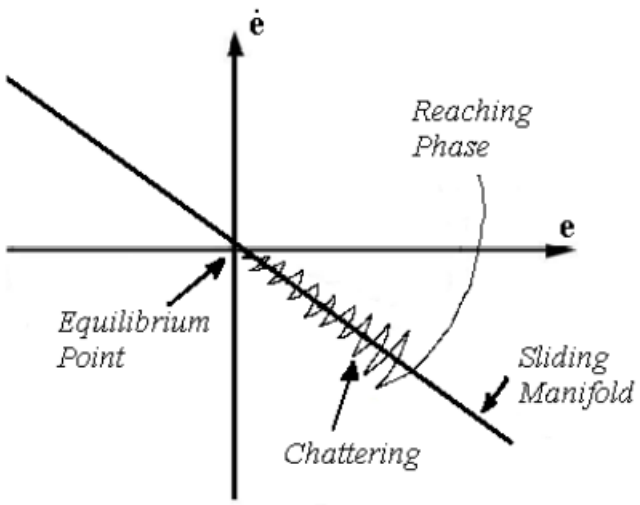


Figure 1.2: Chattering phenomena due to discontinuous term in the control law

A possible solution to overcome this phenomenon is to use another function that can approximate the *sign* function and avoid discontinuity. Typical choices are:

- $\sigma(\eta s) = \frac{\eta s}{1+\eta s}$
- $\sigma(\eta s) = \tanh(\eta s)$
- $\sigma(\eta s) = sat(\eta s)$

1.1.3 Example

Consider the dynamic of a pendulum defined by the following equations

$$\ddot{x} = g\sin(\theta) + d(t) + u$$

, where g=9.81 is the gravity constant, θ is the angle of the pendulum, $d(t)$ is an external disturbance as $2\sin(t)$, and u is the control law to be designed.

Defining the *tracking error* as

$$e = x - x_{des}$$

, it is possible to define the second-order sliding surface as

$$s = \dot{e} + ke$$

,where k is a parameter to be chosen.

Deriving the previous equation is it possible to obtain the first term of the control law u_{eq}

$$\begin{aligned} \dot{s} &= \ddot{e} + k * \dot{e} \\ \dot{s} &= \ddot{x} + k\dot{e}, \quad x_{des} = 0 \\ \dot{s} &= g\sin(\theta) + u_{eq} + k\dot{e} \end{aligned}$$

By imposing $\ddot{s} = 0$, one can obtain

$$u_{eq} = -g\sin(\theta) - k\dot{e} \tag{1.14}$$

The second term of the control law is defined by

$$v = -\rho sign(s) \tag{1.15}$$

, where ρ must be greater than the upper and lower limit of the bounded disturbance, in this case, $L = 2$.

The whole control is defined as

$$u = -g\sin(\theta) - k\dot{e} - \rho sign(s) \tag{1.16}$$

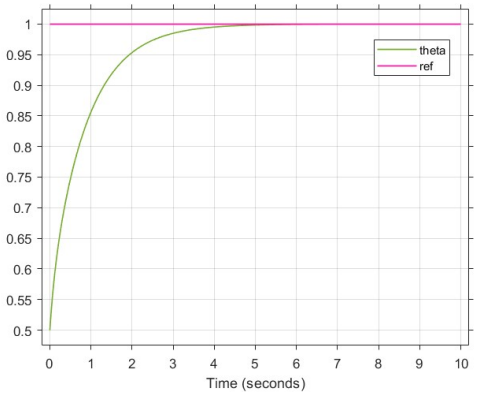
, according to eq.1.8.

The results of the simulations are shown in the following figures.

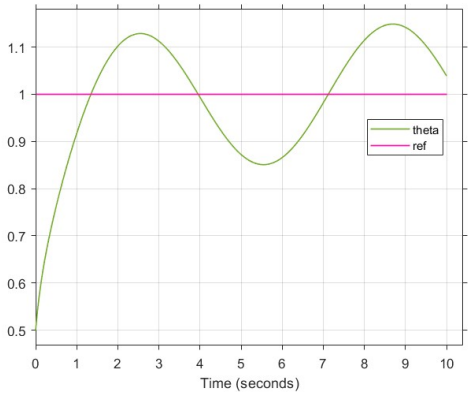
In Figure 1.3, it is possible to notice that simple state feedback is not enough to deal with external disturbance, while the tracking is good when they are not present.

In Figure 1.4, it is possible to notice that eq.1.13 must be satisfied in order to reject the disturbance, in this case, indeed, $\rho < L$.

In Figure 1.5 and 1.6, the sinusoidal disturbance is properly rejected, but the chattering phenomenon can be clearly seen in the control law and sliding surface. When the approximation function $tanh(s)$ is used, the tracking performance is lower but still acceptable and the chattering is no more present.

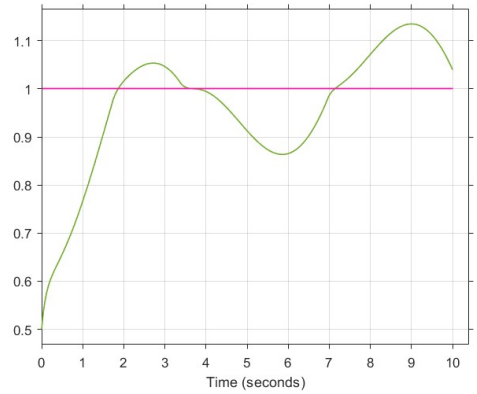


(a) θ when no disturbance is applied

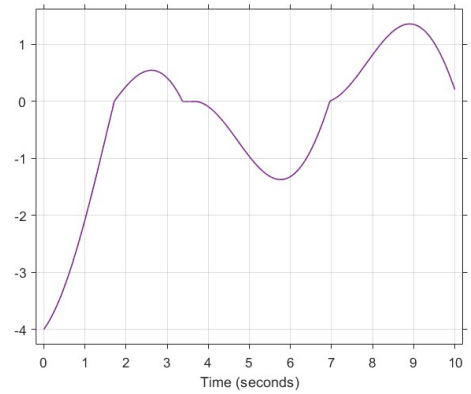


(b) θ when a sinusoidal disturbance is applied.

Figure 1.3: Pendulum response whether a disturbance is applied or not.



(a) θ with ρ=1.



(b) s with ρ=1.

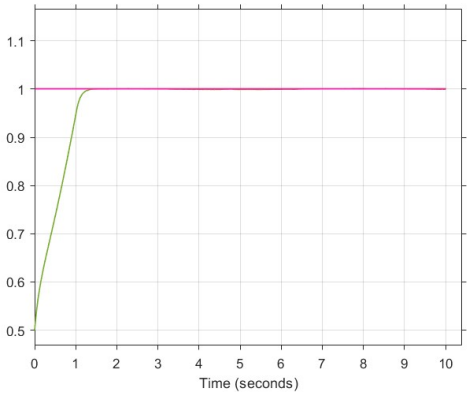
Figure 1.4: Pendulum response with $\rho = 1$,$L = 2$.

1.2 Multi-Agent System

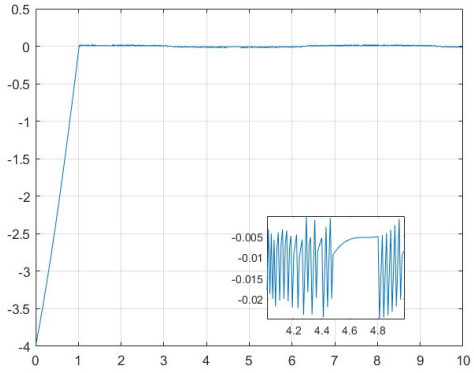
The definition of *Agent* is not commonly shared among the scientific world since an agent can be present in many communities and have several forms, from robots to software, leading to the impossibility to generalize its definition, [24].

The concept of a multi-agent system (MAS) is easier to define since can be applied to all the possible agents interacting with each other. Each agent in a MAS is an intelligent entity capable of perceiving its environment, making decisions, and taking actions based on its internal state and interactions with other agents, with the purpose of obtaining a common goal.

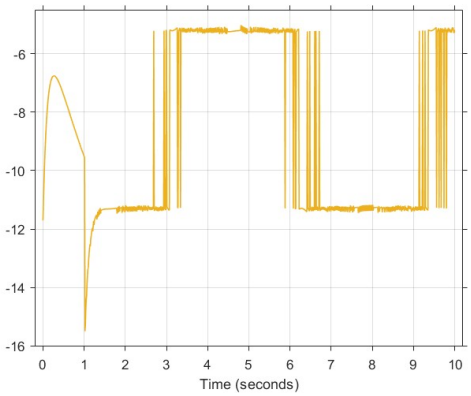
The concept of a MAS draws inspiration from social systems, where individuals interact and collaborate to achieve common objectives. In a MAS, agents can be designed to possess different capabilities, knowledge, and goals, enabling them to perform specialized



(a) θ with $\rho = 3$,$L = 2$.



(b) sliding surface with $\rho = 3$„$L = 2$.



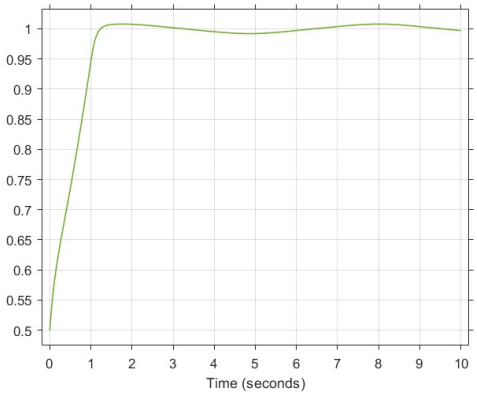
(c) control input with $\rho = 3$„$L = 2$.

Figure 1.5: Pendulum response with disturbance and $sign(s)$.

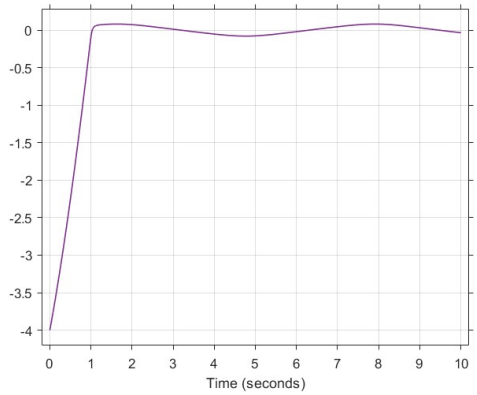
tasks and contribute to the overall system's objectives.

The structure of MAS is said to be *Homogeneous* if all the agents are of the same type and have the same internal architecture. The only difference is its physical location in the space, [25]. An *Hetherogeneous* structure is closer to real-world application, the agents can differ in structure, local goals and their action can depend on the specific location in the environment, [26].

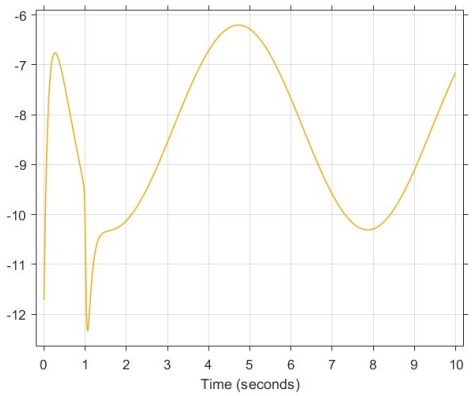
MAS finds applications in various domains, including robotics, distributed systems, social networks, transportation, and economics. In robotics, for example, MAS can be used to coordinate multiple robots, such as consensus and formation control for a fleet of drones [27,28]. In these approaches, a typical choice is to use an *hierarchical organization* with *simple hierarchy*, i.e. a single agent or *virtual leader* [29] is in the highest level and provides information to other branches of the hierarchy tree, [24].



(a) θ with $\rho = 3$„$L = 2$ and $\tanh s$.



(b) sliding surface with $\rho = 3$„$L = 2$ and $\tanh s$.



(c) control input with $\rho = 3$„$L = 2$ and $\tanh s$.

Figure 1.6: Pendulum response with disturbance and $\tanh s$.

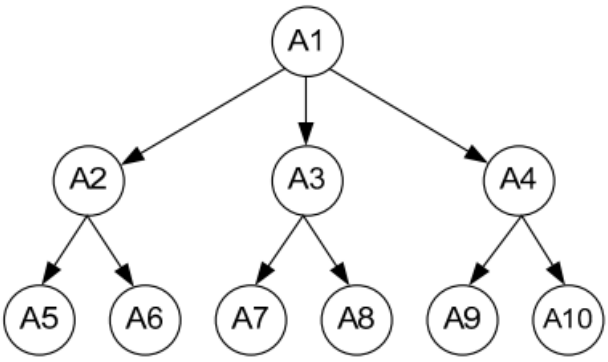


Figure 1.7: A Hierarchical Agent Architecture

1.2.1 Graph Theory

As stated in the previous section, communication and information exchange are very important features in a distributed system. The *Graph Theory* is a useful tool that can

describe the state of the agents and information flow in a multi-agent system. Considering n agents, a graph can be represented as $G = (V, E)$, where $V = 1,2,...,n$ is the number of *vertices* or *nodes*, and corresponds to the number of agents in the graph; $E = (i, j), \quad \forall (i, j) \in V$ is the number of *edges* connecting vertices, and represents the information path between agents. A graph can be *directed*, 1.8a, or *undirected*, 1.8b. In the former case, the edges have orientation and information flows in a specific path and direction, while in the latter case, it can flow in both the direction on the edge between two connected agents.

(a) (b)

Figure 1.8: Directed and undirected graph.

The distribution of information in a graph is strictly related to the notion of *Laplacian* matrix

Definition 2 *A is the Adjacency matrix $A = a(i, j)$, where*

$$a(i,j) = \begin{cases} 1, & (i,j) \in E \\ 0, & otherwise \end{cases}$$

The *Adyancency matrix* can be seen as *diffusion operator*, related to the conservation of the mass in a node, [30].

Definition 3 *D is the Degree matrix, which is diagonal with elements*

$$d(i,i) = \sum_{j=1}^{n} a(i,j)$$

The matrix D gives information on the number of edges linked with a vertex.

Definition 4 *The Laplacian matrix L is defined as*

$$L = D - A$$

In this matrix, most of the information about the structural representation of a graph is hidden within it. In the context of controlling a fleet of drones, the Laplacian matrix can be used to model the connectivity relationships among the drones.

It is constructed such that each row and column represents a drone in the fleet. The elements of the matrix indicate the strength of the connection between the drones. For example, if two drones can directly communicate and exchange information, the corresponding element in the Laplacian matrix will be non-zero. Conversely, if the drones are not directly connected, the element will be zero.

Once the Laplacian matrix is constructed, it can be used to coordinate the drones within the fleet. For instance, distributed control algorithms can be implemented that utilize the Laplacian matrix to exchange information among drones and make collaborative decisions.

In a *Hierarchical* organization, the agents follow a so-called *virtual leader* which is the first to provide information about the common objective, and then this information propagates according to the matrix L.

In this specific organization, not all the agents receive updates from the leader, and so the *Laplacian* matrix should be changed, according to the communication agent-leader.

A new matrix B is defined as a diagonal matrix with element

$$b(i) = \begin{cases} 1, & \text{there is } \textit{agent-leader} \text{ communication} \\ 0, & \text{otherwise} \end{cases}$$

The final matrix,containg all the type of information about communication in the *Multi-Agent system* with *Hyerarchical* organization, is defined as

$$H = L + B$$

1.2.2 Example

Considering the Multi-Agent represented in the next Figure, with $n = 4$ agents and a virtual leader. The agents 1 and 2 receive information directly from the leader and then they update their *neighbors* , respectively agent 3,4 and 3.

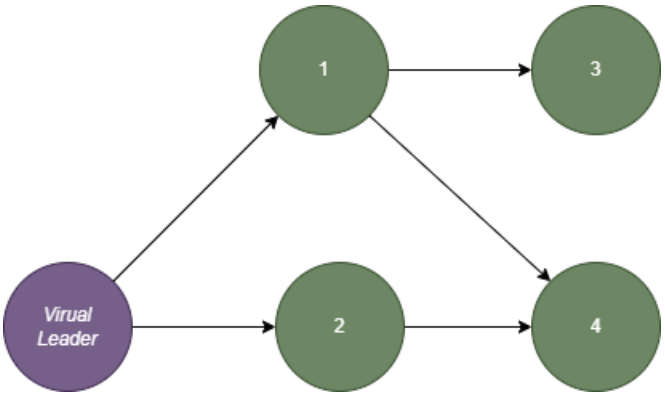


Figure 1.9: Example of *Hierarchical* Multi-Agent system.

In this scenario,

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}$$

.

1.3 Neural Network

The first approach to *Neural Network* dates back to 1943 when McCulloch and Pitts provided a computation model inspired by the human brain neurons, but it is only 40 years later that this topic gained success among researchers thanks to the introduction of the *error back-propagation* method, [31].

Nowadays, Artificial Neural Networks are widely used in all the branches of the industry, solving several problems in real-world applications, such as pattern recognition, speech and voice recognition, automatic translation, and so on.

In the engineering control field, several applications of the *Neural Adaptive* control have been developed to control a quadrotor [32, 33], a robot manipulator [34], and also in the electrical field for an electromagnetic suspension system [35].

1.3.1 Computational model

Neurons in the human brain continuously receive a large number of inputs from the external world but not all of them have our attention. This is because a neuron is *normally closed*, i.e. it must be activated to transmit the received input, and so a certain *activation treshold* must be taken into account to model this behavior. Moreover, not all the inputs have the same *weigths*, which means that we can give a different value to different inputs; for example, our brain tries to reject white noise while we are focused or we are listening to music.

In the last years, the success of neural networks in estimating functions between input and output led to the development of different configurations and thus with specific characteristics.

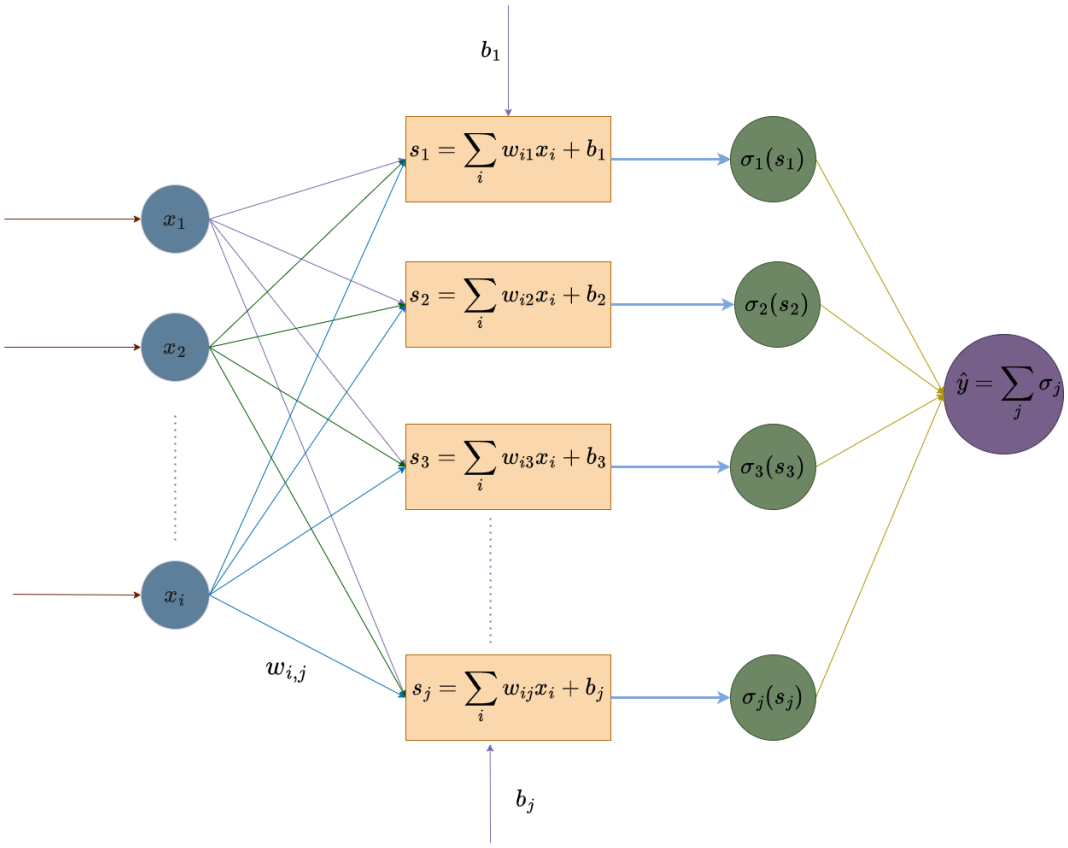


Figure 1.10: Artificial Neural Network model.

The *single-layer Perceptron* is the easiest one where the inputs are directly linked to the output and thus can be applied to simple cases and in linear problems. To increase its potential and use the neural network for complex problems and learn non-linear relations between data, the *Multi-layer Perceptron (MLP)* has been modeled and it is the one presented in Figure 1.10. Neural networks can also have a memory that can be short-term in *Recurrent Neural Network* or *LongShort-Term Memory*.

In the following, the MLP with a single hidden layer will be used in this thesis to learn to estimate the function linking the input data to the outputs.

The concept of neural networks might sound complex and difficult but the tasks of a single neuron are trivial: propagate the received input to the neighbors and update(*learning*) its weights.

In Figure 1.10, it is possible to identify a configuration that is in common among all the possible ones: from left to right,*input layer*,*hidden layer*, and *output layer*. In the first one, the inputs can arrive from other neighbors or an external source. The *learning* phase is related to the hidden layer whose neurons or *nodes* produce outputs that are equal to the weighted sum of the inputs coming from the previous layer. The output from these neurons is a value that depends on an activation function whose input is the result of the weighted sum. The neuron is said to be active if the weighted sum is higher than

a certain threshold.

Each neuron of the hidden layer receives all the inputs in it and then it sums it according to the following equation

$$s_j = \sum_i w_{ij} x_i + b_j \tag{1.17}$$

, where x_i is the i_{th} input, w_{ij} is the weight linking the input i to the neuron j, and b_j is the treshold or *bias* for the activation function, which is

$$\sigma(s_j)$$

and σ can be

- $\sigma_j = sgn(s_j)$
- $\sigma_j = \frac{1}{1+e^- s_j}$
- $\sigma_j = ks_j + b$

The predicted output, which should be as close as possible to the actual one, can then be related to the inputs as

$$\hat{y} = \sum_j \sigma(s_j)$$

$$\hat{y} = \sum_j \left(\sigma(\sum_i w_{ij} x_i + b_j) \right) \tag{1.18}$$

1.3.2 Radial Basis Function

Another type of activation function is the so-called *Radial Basis Function*, which was introduced in 1985 by M.J.D. Powell [36], but the first application is by Broomhead and Lowe, [37].

The most used RBF function is the *Gaussian* function defined as

$$\phi(x) = e^{\frac{\|x - c\|}{\sigma^2}}$$

, where x is the input, c is the center of the symmetric of the Gaussian function and σ is the width of the basis function.

The center c can be thought to be the *mean value* of the input x, while σ shapes the bell function. The influence of these parameters on the RBF is reported in the next figure.

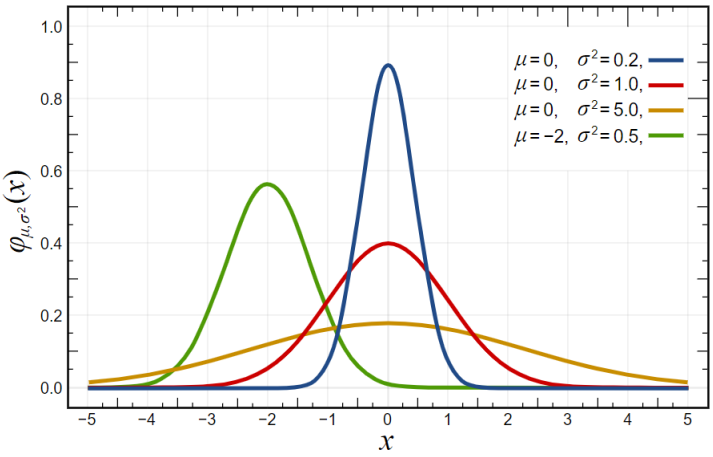


Figure 1.11: Influence of c and σ on the *Gaussian* function.

1.3.3 Multi-layer with error back-propagation

The estimation process with neural networks relies on the *learning rule*, i.e. updating the weights connecting the inputs to the hidden layer in order to minimize the difference between the desired output and the one coming out from the neural network. The learning rule is indeed an *optimization* process and it can be described by a few steps

1. Choose a random initial value for the weight w_{ij}.
2. Provide the input to the hidden layer and computes the estimated output $\hat{y}$.
3. If $\hat{y} \neq y$ then the weights must be updated,Δw_{ij}, according to a certain policy.
4. Repeat until the *estimation* error, $e = y - \hat{y}$, is close to zero.

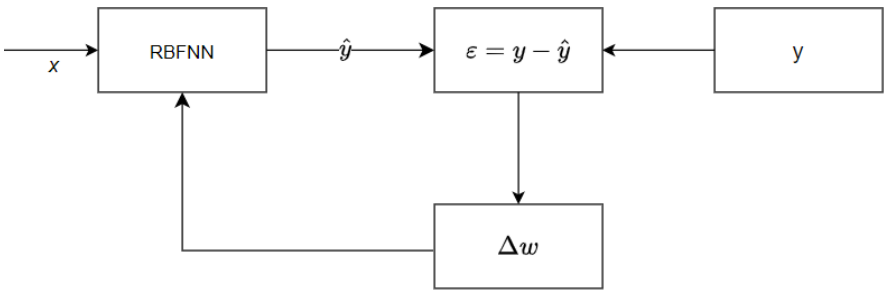


Figure 1.12: Scheme of the *learning* process.

The updated value of the weight can be described by

$$w_i(t+1) = w_i(t) + \Delta w_i \tag{1.19}$$

where the variation can be found through an optimization policy(*back-propagation*) that should drive w_i towards an unknown value w^* such that it minimizes a *cost function* E

$$w^* = \underset{w}{argmin}\ E(w)$$

and the cost function is defined as

$$E = \frac{1}{2}(y - \hat{y})^2 \tag{1.20}$$

$$\hat{y} = \sum_j \sigma_j(s_j) \tag{1.21}$$

$$s_j = \sum_i (w_{ij}x_i + b_j) \tag{1.22}$$

For *complex* optimization problem, finding a *global* minimum is easier compared to a *non-complex* problem, since in the latter case, the solution could converge to a *local* minimum, ignoring other possible solution.

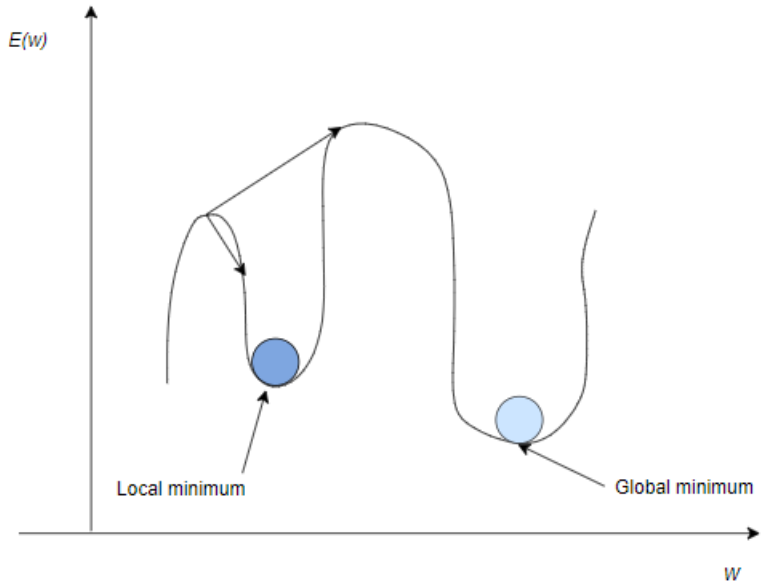


Figure 1.13: *Local* and *Global* minimum.

To avoid the algorithm of the learning rule stopping in a local minimum, it is important to introduce the *learning rate*, γ, which describes the width of next the next step.

The updating term in eq.1.19 can be described by

$$\Delta w = -\gamma \frac{\partial E}{\partial w} \tag{1.23}$$

The meaning of this equation is straightforward: the two terms are discordant and so whenever the cost function is increasing, the weights should decrease their values to counteract the misleading behavior, and vice versa.

Taking into consideration the scheme in Figure1.10 and eq.2.10, using the *chain rule* once can write

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial s} \frac{\partial s}{\partial w} \tag{1.24}$$

$$\frac{\partial E}{\partial \hat{y}} = -(y - \hat{y}) \tag{1.25}$$

$$\frac{\partial \hat{y}}{\partial s} = \sigma'(s) \tag{1.26}$$

$$\frac{\partial s}{\partial w} = x \tag{1.27}$$

If one defines $\varepsilon = (y - \hat{y})$, it is possible to rewrite eq.1.24 as

$$\frac{\partial E}{\partial w} = -\varepsilon \sigma'(s) x \tag{1.28}$$

$$\Delta w = \gamma \varepsilon \sigma'(s) x \tag{1.29}$$

Although *back-propagation* can be applied to a network with any number of layers, *the universal approximation theorem* states that a single-hidden-layer suffices to approximate most of the function if the activation function is a non-linear one, in this thesis the *RBF*.

1.3.4 Considerations and Drawbacks of the back-propagation

The neural networks approach requires choosing several parameters, number of *neurons*, *activation function* and *treshold*, number of *layers*, *learning rate*, number of *input samples* and each one can affect the quality of the entire system.

Considering *sigmoid* as an activation function, for example, can drive the system to a phenomenon called *network paralysis*.

The weights can reach high absolute values, and so the activation function can converge in a boolean one, with an output value that can be only *on-off*, losing all the range in between. The MLP neural network, thus, becomes a single-layer network with worse performances.

Another issue can be related to the learning rate, where a too-small value will bring the system to the optimal solution but in a quite long time. Furthermore, the algorithm can not be willing to look for other solutions once he already found a minimum, so consider a local minimum as a global one. A too-large value can lead to missing some solutions as well but the system is more willing to search for a new and maybe better one.

Small values of learning samples can fool the system. The learning process stops when the estimation error is equal to zero. A too-small set of input, thus, can bring the neural network to approximate a function that is not the desired one, but the network is blind to this issue since its performance parameters are equal to zero.

Another issue can arise due to the number of hidden layers. A high number corresponds to higher intelligence, but this is not always good. Indeed, a too-smart network can also learn what it is not supposed to learn, the disturbances, [38].

Chapter 2

LIMBIC System

2.1 Biological Model

The Limbic system is a complex and fascinating network, responsible mainly for creating new memories and related emotions, needed to regulate behavior and, in some cases, even for the survival of the species.

Indeed, its presence can be seen even in branches of the phylogenetic tree where animals have a rhinencephalon brain, i.e. the brain is directly connected with the nose, which is simplified compared to human beings.

In the rabbit, as an example, the limbic system helps to correctly distinguish between a predator, a potential partner, and food. This is because the inputs coming from the olfactory apparatus are associated with emotions, which are a result of the experience developed through its lifetime, and so of the memory linked to perceived inputs. All these functions (perception, memory, emotion) are part of a more complex process, the learning process.

2.1.1 Components of the Limbic System

Over the course of evolution, the limbic system in the human brain has undergone complexity, yet its fundamental purpose of learning remains unaltered. While the limbic system has expanded and encompassed a greater number of components, its primary functions persist albeit with certain refinements and adaptations. This intricate system has evolved to process a diverse range of signals and stimuli, employing an expanded repertoire of structures to fulfill its core functions:

- **Sensory Cortex**: also known as the primary sensory cortex or primary sensory areas, is a crucial component of the limbic system that plays a fundamental role in processing and interpreting sensory information from the external environment. Located in different regions of the brain, the sensory cortex consists of specialized areas dedicated to specific sensory modalities, including vision, hearing, touch, taste, and smell.

- **Amygdala**: consisting of several nuclei, is a key component of the limbic system.

It plays a pivotal role in emotional processing and fear conditioning. The amygdala receives sensory information from various sources, including the thalamus, and rapidly evaluates the emotional significance of stimuli. It also helps to encode emotional memories and plays a role in emotional responses and decision-making.

- **Hippocampus**: located in the medial temporal lobe, is crucial for memory formation and spatial navigation. It acts as a temporary storage site for short-term memories and facilitates their conversion into long-term memories through a process known as consolidation. The hippocampus also plays a role in contextual memory and is essential for episodic memory, allowing us to recall specific events from our past.

- **Thalamus**: it serves as a relay center for sensory information, receiving inputs from various sensory systems and routing them to the appropriate cortical areas. In the context of the limbic system, the thalamus plays a vital role in relaying sensory information to the amygdala and other limbic structures, allowing for the emotional evaluation of stimuli.

- **Hypothalamus**: situated below the thalamus, the hypothalamus influences emotional responses and behaviors through its connections with the amygdala and other limbic regions. It also regulates hunger, thirst, and body temperature.

- **Orbitofrontal Cortex** : located in the frontal lobes just above the eyes, it receives inputs from various limbic regions, including the amygdala, and integrates emotional and cognitive information to guide behavior. The orbitofrontal cortex plays a crucial role in assigning emotional value to stimuli, evaluating rewards and punishments, and inhibiting inappropriate or maladaptive responses.

In this thesis, a simplified model containing only Sensory Cortex, Thalamus, Amygdala, and Orbitofrontal Cortex will be considered, and it is shown in Figure 2.1.

2.2 Computational Model

In its book, Mowrer [1], has shown how emotions play an important role in the so-called emotional learning two-steps process and that it can be easily implemented.

The input signals of the Amygdala come from the Sensory Cortex and the Thalamus, and they have to be previously paired with a reinforcer signal, which nature is still unknown. In the learning process, nuclei of the Amygdala react to the stimuli only after the association with their emotional consequences has been created, and so the link between the stimulus and its response. [39] [40]

Even if the main part of the learning process is taken in the Amygdala, the role of the Orbitofrontal Cortex is very crucial. Indeed, it helps the Amygdala to inhibit associations that are no longer valid and to detect the omission of reinforcement.

It only receives signals from the Sensory Cortex, then compares the expected output with the actual one, and inhibits learning in the Amygdala based on this difference. [3]

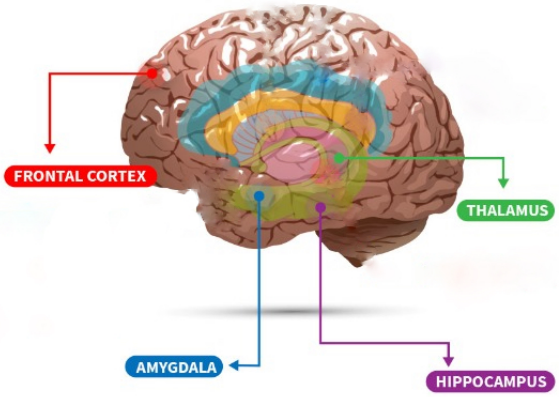


Figure 2.1: Limbic System.

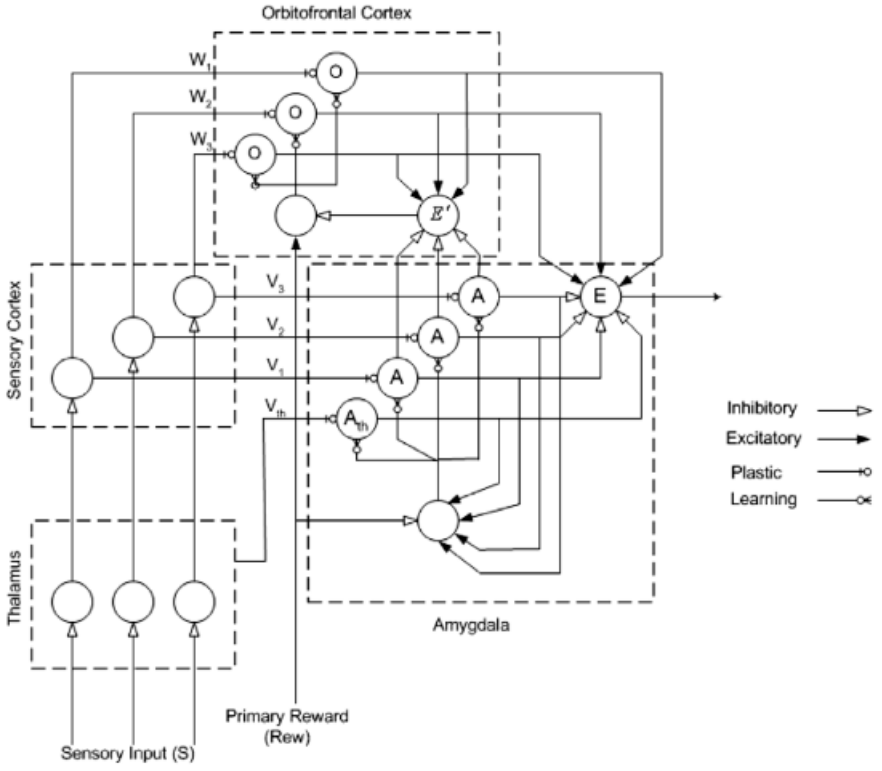


Figure 2.2: Simplified model of the Limbic System: Thalamus, Sensory Cortex, Orbitofrontal Cortex, and Amygdala.

As shown in Figure 2.2, [3], the inputs for *Amygdala* and *Orbitofrontal Cortex* come from the *Cortex* area: the main contribution of the *Thalamus* is to pre-process signals

and then inform the *Amygdala* of their presence, [3]. Once the *Sensory Signals (S)* is processed, they enter the *Sensory Cortex* area for a finer analysis.

2.2.1 Amygdala

The *Amygdala* has a neuron for each input coming from the *Cortex* area and is characterized by a plastic connection with a specific weight:

V_i for the $i-th$ signal coming from the *Sensory Cortex* and V_{th} for the input coming from the *Thalamus*.

Each neuron, or node, produces an output as a result of multiplication between the weight and its respective signal:

$$A_i = S_i * V_i \tag{2.1}$$

Then, the outputs of the *Amygdala* are:

$$A = \sum_j S_i * V_i; \tag{2.2}$$

$$A_{th} = \max(S_i) * V_{th}; \tag{2.3}$$

As shown in Figure **??**, these outputs are feedback in a specific node where a comparison with the external *reward* is evaluated. The learning process is basically based on the updates of the weights according to mismatches between the outputs of the *Amygdala* and the *Rewards*.

The plastic connections are updated according to the following equation:

$$V_i(t) = V_i(t-1) + \delta V_i \tag{2.4}$$

where the variation is calculated as:

$$\delta V_i = \alpha * S_i * \max(0, Rew - \sum_j A_j) \tag{2.5}$$

The most important characteristic of the 2.5 is that is a monotonic function and so the weights of the *Amygdala* can not decrease, i.e. this area can not forget and it is only able to learn. [3]

The inhibition process is a task of the *Orbitofrontal Cortex*.

2.2.2 Orbitofrontal Cortex

The structure of the *Orbitofrontal Cortex*, herein forth indicated as *OFC*, is almost the same as the one of the *Amygdala*. In this area, though, the information from *Thalamus* is not visible.

It has a node for each input signal coming from the *Sensory Cortex*, and it also receives as input the output coming from the *Amygdala*, but without considering the *Thalamus*.

The neurons are characterized by a plastic connection with weight W_i and they produce an output as a multiplication between the wight and its respective signal:

$$O_i = S_i * W_i \tag{2.6}$$

In this case, the learning process is evaluated as a comparison between the expected and actual output:

$$W_i(t) = W_i(t-1) + \delta W_i \tag{2.7}$$

where the variation is calculated as:

$$\delta W i = \beta * S_i * (E' - Rew) \tag{2.8}$$

β is the learning rate, S_i is the input signal and

$$E' = \sum_j A_j - \sum_j O_j \tag{2.9}$$

The crucial difference between *Amygdala* and *OFC* can be found in analyzing the learning or updating rule for their plastic connections.

The weight W_i can increase and decrease, a behavior that is needed to emulate the inhibition process, while V_i has a monotonic increasing behavior, emulating the learning process throughout the experience. [3]

To summarize, both subsystems receive the same input signals, except for the *Thalamus*. Each subsystem processes the signal according to its neurons and weights and provides an output that is feedback to generate the updating law.

The final output of the global system is generated as:

$$E = \sum_j A_j + A_{th} - \sum_j O_j \tag{2.10}$$

.

2.2.3 Results of the *Acquisition* process

In the year 2000, Morén and Balkenius published "A Computational Model of Emotional Learning in the Amygdala" [3], an article aiming to propose a preliminary computational model of the Limbic system, with more focus on *Amygdala* and *OFC*.

They run a set of simulations to verify this model and, in particular, the learning process also called *acquisition*. Their result is proposed in Figure 2.3:

When *Reward* and the input signals are present at the same time, the output E begins to track the desired *Reward*, with a certain reaching time. The *Amygdala* and *Thalamus* change the weights of their neural networks since they are reacting to the presence of the *Reward* and thus they are undergoing the learning process.

Once, the output E reaches the steady state, the V and V_{th} are constants, meaning they have learned the signals in input.

Around 300 s, the *Reward* is not present anymore. As we can see, the output E drops to zero, as a consequence of the inhibition process led by the *OFC*, and the increasing value of the weights W.

The weights of the *Amygdala* are not affected and their values are equal to the previous ones when the *Reward* was present.

The *Reward* appears again, around 600 seconds, and the reaching time of the output E is smaller compared to the first scenario when the system was still learning. This is

because the system has past experience with the input signal and thus it already knows how to react. Also, V_{th} and V do not change since the signal to be tracked has no new pieces of information with respect to the first one. [3]

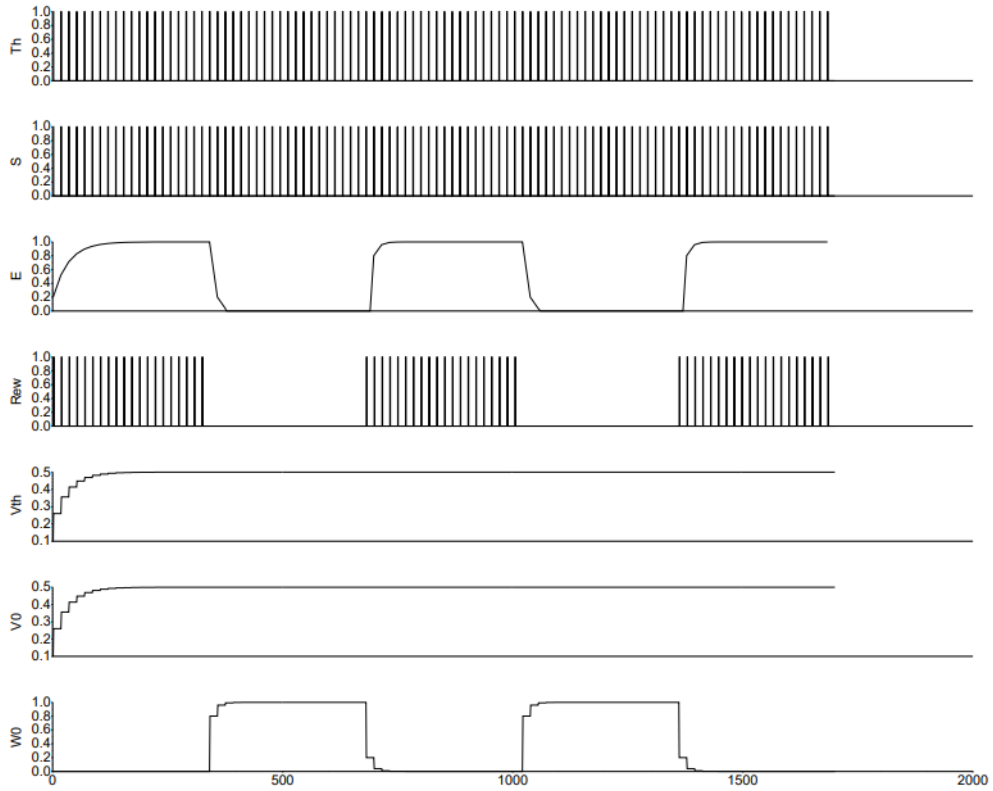


Figure 2.3: Results of the *Acquisition* process. From up to bottom: Thalamus, Sensory Signal, Output E, Reward, V_{th}, V, W.

In Figure 2.4, it is pictured the difference in whether the Orbitofrontal Cortex is inhibiting or not the learning process in the Amygdala. As one can see when the Orbitofrontal Cortex is not active and the Reward is not present, the Amygdala is not capable to unlearn the input signal and it decreases very slowly compared to when the inhibition process is called by the Orbitofrontal Cortex; it is said that the extinction is a passive process.

2.3 Reward and Sensory Signal

A crucial aspect in the design of the BELBIC is the selection of the Reward and Sensory signals, which play a vital role in capturing the current state of the system.

In the existing literature, it is common to represent the Sensory signal in the form of a Proportional-Integral-Derivative (PID) or a Linear-Quadratic Regulator (LQR). This choice allows for a more intuitive interpretation of the Sensory signal, taking advantage of well-established control techniques that have been developed in recent years, [41].

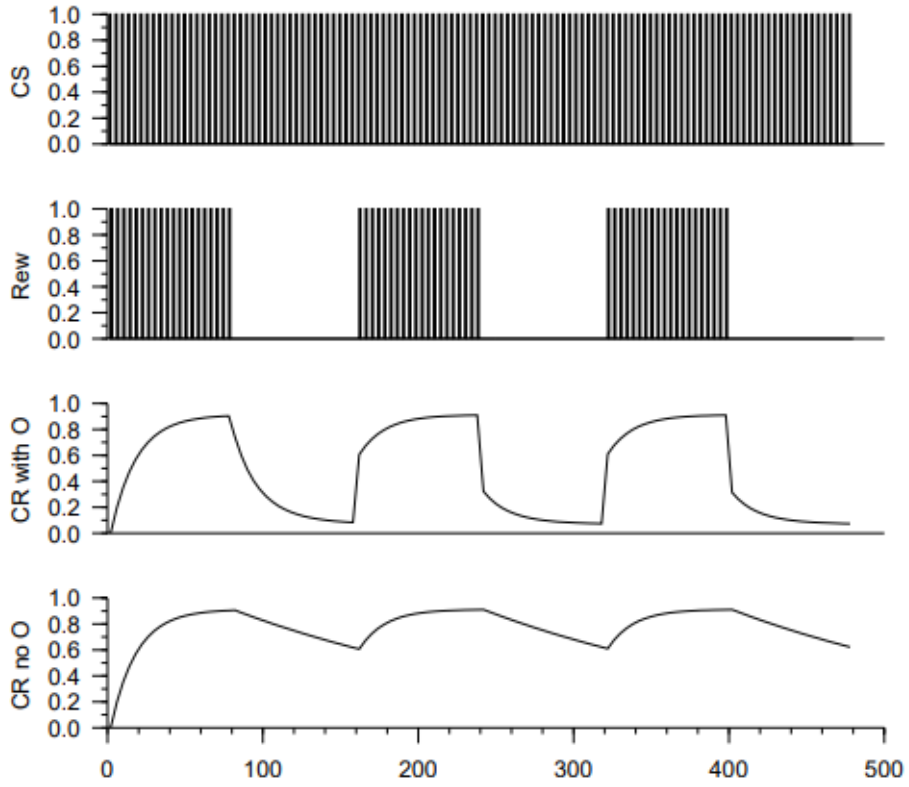


Figure 2.4: Conditioned Response(CR) and Unconditioned Response.

By employing a PID or LQR-based Sensory signal, the BELBIC framework can effectively integrate emotional and cognitive elements into the control process, enhancing its adaptability and responsiveness to system dynamics. This design choice ensures that the Sensory signal provides meaningful information about the system's current conditions, enabling the BELBIC to make informed and effective control decisions.

The selection of a PID or LQR-based Sensory Signal is not obligatory, and the design approach allows for flexibility and arbitrariness. In fact, any type of function can be considered a suitable candidate for these signals, [42].

In the general approach of BELBIC, the Sensory Signal is typically formulated as functions of the plant's states and outputs. This allows for capturing relevant information about the system's behavior and performance. On the other hand, the Reward Signal is constructed as a function of both the plant and controller outputs.

$$S = f(r, x_1, \ldots, x_n, y_1, \ldots, y_m) \tag{2.11}$$

$$Rew = g(e_1, \ldots, e_n, u_1, \ldots, u_m) \tag{2.12}$$

The overall control scheme is depicted in the following Figure 2.5. The RSG block is the *Reference Signal Generator* which receives as input the tracking error, the control input, and actual states and provides as output the result of eq.2.11 and eq.2.12. These

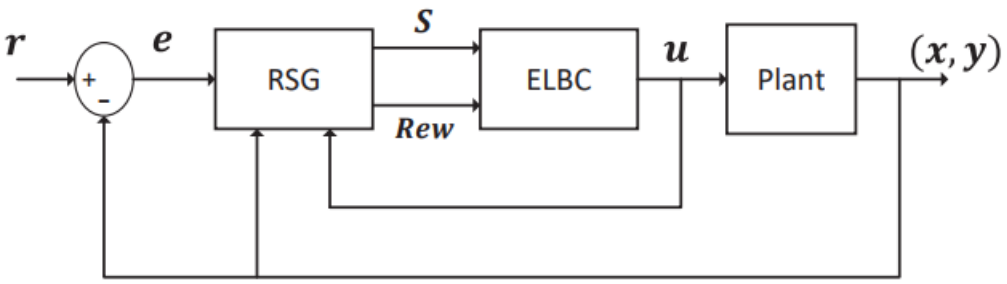


Figure 2.5: Control scheme with BELBIC.

signals enter the *Emotional Learning- Based Controller* block, whose scheme can be seen in Figure 2.2, and the control signal can be established according to eq.2.10, which is the result of the learning process undergone in the Amygdala and Orbitofrontal Cortex.

It is possible to implement the previous scheme in MATLAB/Simulink in the following way

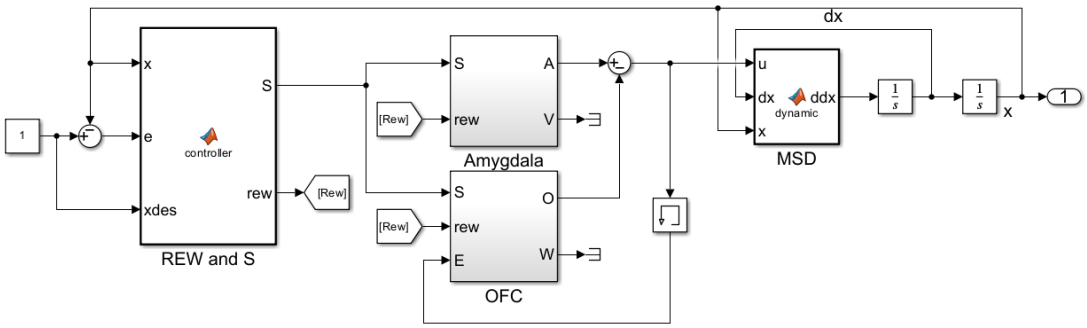
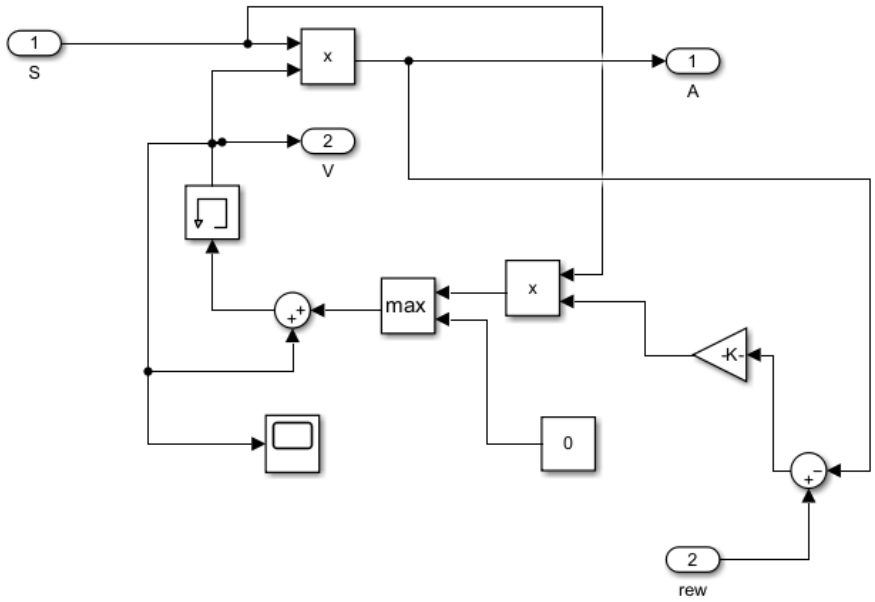
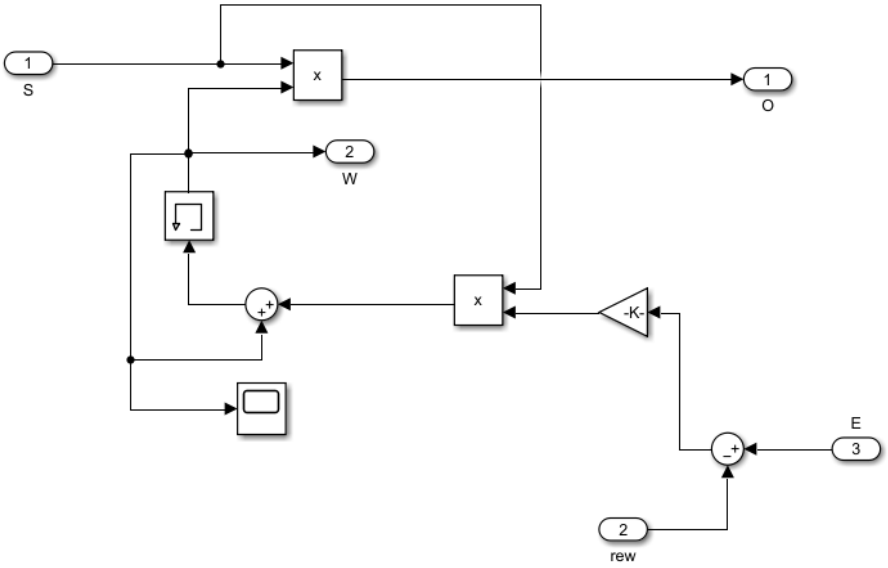


Figure 2.6: Simulink Implementation of the control scheme.

The Amygdala and OFC block are shown in the following figures, [43–45].



(a) Amygdala.



(b) Orbitofrontal Cortex.

Figure 2.7: Simulink scheme for Amygdala and OFC.

Chapter 3

Neural Adaptive SMC vs NFTSMC

In the following sections an overview of new advanced typologies of *sliding surface* will be analyzed and compared to the classical one described in Chapter 1. New terms are added to the simple sliding surface, analyzing possible *singularities* that occur in the *control law* and how this issue has been solved in the literature, [46].

Finally, *Non-Singular Fast Terminal Sliding Surface*(NFTSMC) is used with a neural adaptive control, and a numerical simulation is provided and compared with the same type of control typology but with SMC.

3.1 Terminal Sliding Mode Control

This new type of sliding mode control was introduced for the first time in 1992, [47], used initially for the SISO system, [48], and then extended to the MIMO system, [49].

Its advantage relies on the concept of *terminal attractor* introduced in [50], which leads to the concept of *terminal slider*.

3.1.1 Terminal attractor

In the classic dynamic, the existence of a unique solution subject to fixed initial conditions plays a crucial role in solving the differential equations describing the system.

A system has a unique solution only if it satisfies the *Lipschitz* condition.

Consider a classic dynamic system,

$$\dot{x} = f(x_1, x_2, \dots, x_n)$$

, the *Lipschitz* condition is defined as

$$\left| \frac{\partial f_i}{\partial x_j} \right| < \infty$$

which means that partial derivatives of the systems must be bounded.

Classical dynamics is not coherent if applied to some particular systems that are irreversible and deterministic in real life such as, for example, thermodynamics, where most of the processes are irreversible.

To describe this type of system a new form of differential equation should be considered,

$$\dot{x} = f^{q/p}(x_1, x_2, \ldots, x_n)$$

, where q, p are odd integers and $q < p$. Therefore,

$$\left|\frac{\partial \dot{x}_i}{\partial x_j}\right| = \frac{q}{p}(f)^{q/p-1}(x_1, x_2, \ldots, x_n)\left|\frac{\partial f_i}{\partial x_i}\right| \to \infty \quad \text{, if } x_i \to 0.$$

, [51], which clearly does not satisfy the necessary condition for the uniqueness of the solution, but the equilibrium points are *terminal attractor*, where the term *terminal* refers to finite-time convergence toward the desired state, which reflects a more realistic behavior in real-world applications.

The finite-time convergence can be enhanced by comparing the simplest dynamical system with a terminal one.

Consider the following systems,

$$\dot{x} = -x \tag{3.1}$$

$$\dot{x} = -x^{\frac{1}{3}} \tag{3.2}$$

, the solutions to these differential equations are

$$\int_{x(0)}^{x} \frac{dx}{x} = -\int_0^t dt \quad \to x(t) = x(0)e^{-t} \quad \to t = \infty \tag{3.3}$$

$$\int_{x(0)}^{x} \frac{dx}{x^{\frac{1}{3}}} = -\int_0^t dt \quad \to x(t) = x(0) - \frac{2}{3}t^{\frac{3}{2}} \quad \to t = \frac{3}{2}x(0)^{\frac{2}{3}} \tag{3.4}$$

, from which it is possible to calculate the settling time as the time required to steer the equations toward the equilibrium $x = 0$,

From equations 3.4 and 3.3 one can see that in the former case, the convergence time is finite, which defines the equilibrium solution of the system a *terminal attractor*. In the classical system, this is impossible due to its nature, which allows it to reach a steady state in infinite time.

In *Terminal Sliding Mode Control*(TSMC), this concept can be used to perform better than the classical approach.

Again, consider a second-order system of the form

$$\ddot{x} = f(x) + g(x)u + d(x, t)$$

, the control law to drive the system to the desired states can be derived from a sliding surface that contains the terminal attractor

$$s = \dot{x} + \beta|x|^{\lambda} sgn(x) \tag{3.5}$$

, where $x \in \Re$ is a variable, $\beta > 0$, and $0 < \lambda < 1$ are parameters to be tuned, with the latter one defined as $\lambda = q/p$ where q and p are odd integers to guarantee continuity in the derivative and $q < p$.

When $s = 0$ it is possible to define the time needed to reach the desired states

$$\begin{aligned} s &= \dot{x} + \beta |x|^{\lambda} sgn(x) = 0 \\ \dot{x} &= -\beta |x|^{\lambda} \end{aligned} \tag{3.6}$$

Separating the variable and integrating leads to

$$t_s = \beta^{-1}(1-\lambda)^{-1}|x(0)|^{1-\lambda}$$

This settling time can be achieved if the control law is derived from eq.3.6,

$$\dot{s} = \ddot{x} + \lambda\beta|x|^{\lambda-1}\dot{x} = 0 \tag{3.7}$$

$$u_{TSMC} = \frac{1}{g(x)}(-f(x) - \lambda\beta|x|^{\lambda-1}\dot{x} + v) \tag{3.8}$$

, where v is the additional term defined in eq.1.15.

3.2 Fast Terminal Sliding Mode Control

Once the convergence time is finite, it is possible to add a term into the equation of the sliding surface to have a faster convergence, as developed in [52].

The new sliding surface can be defined as

$$s = \dot{x} + \alpha x + \beta |x|^{\lambda} sgn(x) = 0 \tag{3.9}$$

$$\dot{x} = -\alpha x - \beta |x|^{\lambda} sgn(x) \tag{3.10}$$

Consequently, the convergence fast finite time can be computed as

$$t_s = \frac{\ln(\alpha|x(0)|^{1-\lambda} + \beta) - \ln\beta}{\alpha(1-\lambda)}$$

The equation of the sliding surface suggests that when the system is far from the desired states, the new term prevails on the one with a power lower than one and thus it helps the system to move faster toward the sliding surface.

When the system approaches the equilibrium point, the terminal attractor prevails on the linear term and steers faster the system toward the sliding surface.

The control law can be extracted from the derivative of the sliding surface, as usual.

$$\dot{s} = \ddot{x} + \alpha x + \beta |x|^{\lambda} sgn(x) = 0 \tag{3.11}$$

$$u_{FTSMC} = \frac{1}{g(x)}(-f(x) - \alpha x - \lambda\beta|x|^{\lambda-1}\dot{x} + v) \tag{3.12}$$

3.3 Non-Singular Terminal Sliding Mode Control

The previous two sliding mode controls may face a singularity due to the term $-\lambda\beta|x|^{\lambda-1}\dot{x})$, and it occurs when $x = 0$ and $\dot{x} \neq 0$ since the control law tends to infinite. The Non-Singular Terminal Sliding Surface has been developed to avoid this type of problem while guaranteeing the finite convergence time of the system, [53].

The sliding surface is defined as follows

$$s = \beta|\dot{x}|^{\lambda} sgn(\dot{x}) + x = 0 \tag{3.13}$$

, where $\lambda = p/q$ and $1 < \lambda < 2$.

Thus consequently, the convergence time is

$$t_s = \frac{\beta^{\frac{1}{\lambda}}|x(0)|^{1-\frac{1}{\lambda}}}{1-\frac{1}{\lambda}}$$

, which can be achieved by the following control law

$$\dot{s} = \beta\lambda|\dot{x}|^{\lambda-1}\ddot{x} + \dot{x} \tag{3.14}$$

$$u_{NTSMC} = \frac{1}{g(x)}\left(-f(x) - \frac{\dot{x}}{\lambda|\dot{x}|^{\lambda-1}} + v\right) \tag{3.15}$$

$$u_{NTSMC} = \frac{1}{g(x)}\left(-f(x) - \frac{1}{\lambda}|\dot{x}|^{2-\lambda} + v\right) \tag{3.16}$$

3.4 Non-Singular Fast Terminal Sliding Mode Control

This sliding surface merges all the previous ones discussed in the last sections. It is defined as follows, [54]

$$s = x + k_1|x|^{\alpha} sign(x) + k_2|\dot{x}|^{\beta} sign(\dot{x}) \tag{3.17}$$

, where α and β are parameters to be properly choosen.

Consequently, the control law

$$\dot{s} = \dot{x} + k_1\alpha|x|^{\alpha-1}\dot{x} + k_2\beta|\dot{x}|^{\beta-1}\ddot{x} \tag{3.18}$$

$$u_{NFTSMC} = -\dot{x}\left(1 + k_1\alpha|x|^{-1}\right)\left((k_2\beta)^{-1}|\dot{x}|^{1-\beta}\right) + v \tag{3.19}$$

3.5 Example

In this section, all the sliding mode controls described in this chapter are compared by means of a simple example.

Consider the second-order dynamic for a *mass-spring-damper* system in Figure 3.1

$$m\ddot{x} + c\dot{x} + kx + d(t) = F \tag{3.20}$$

$$\ddot{x} = \frac{1}{m}(F - c\dot{x} - kx) \tag{3.21}$$

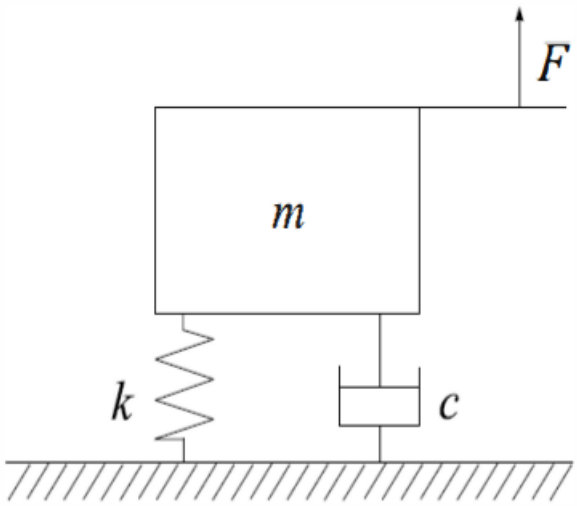


Figure 3.1: Mass-Spring-Damper System.

, where $m = 1, c = 3, k = 2$, and F is the control input;at first external time-dependent disturbance is not considered.

This dynamic is *affine* with the control u and so the sliding mode control can be applied

$$\begin{aligned} \ddot{x} &= f(x) + g(x)u \quad , where \\ f(x) &= \frac{1}{m}(-c\dot{x} - kx) \\ g(x) &= \frac{1}{m} \end{aligned}$$

The goal of the system is to track the input signal x_{des} and thus steer the tracking error to zero, $\lim_{t\to\infty} e = 0, \quad e = x - x_{des}$.

For a classical sliding mode control, it is possible to define the following sliding surface

$$s = \dot{e} + ke$$

, where k is a design parameter. This leads to the control law

$$\begin{aligned} \dot{s} &= \ddot{e} + k\dot{e} = 0 \\ \dot{s} &= \ddot{x} - \ddot{x}_{des} + k\dot{e} = 0 \\ \dot{s} &= f(x) + g(x)u - \ddot{x_{des}} + k\dot{e} = 0 \\ u &= \frac{1}{g(x)}(\ddot{x}_{des} - f(x) - k\dot{e}) \end{aligned}$$

The latter equation shows how the control signal directly depends on the choice of the sliding surface.

Furthermore, a Neural Adaptive approach to identify the functions of the dynamic is applied to estimate $f(x)$ and $g(x)$. This is done because this approach is the closest one to the BELBIC, and thus the results can be compared.

The Neural Adaptive is composed of a multi-layer with one single hidden layer with 50 neurons and RBF as activation functions. The adaptation law is defined by back-propagation, eq.1.23. The outputs of this system are $\hat{f}(x)$ and $\hat{g}(x)$, an estimate of the real dynamic.

The control signal is then modified as follows

$$u = \frac{1}{\hat{g}(x)}(\ddot{x}_{des} - \hat{f}(x) - k\dot{e})$$

, and it is modified for each sliding surface discussed previously.

Results of the tracking are compared using two typical performance parameters, the Integral Absolute value of the Error, IAE, and the Integral Square Value of the control signal, ISV.

$$IAE = \int_0^t |e|dt$$
$$ISV = \int_0^t u^2 dt$$

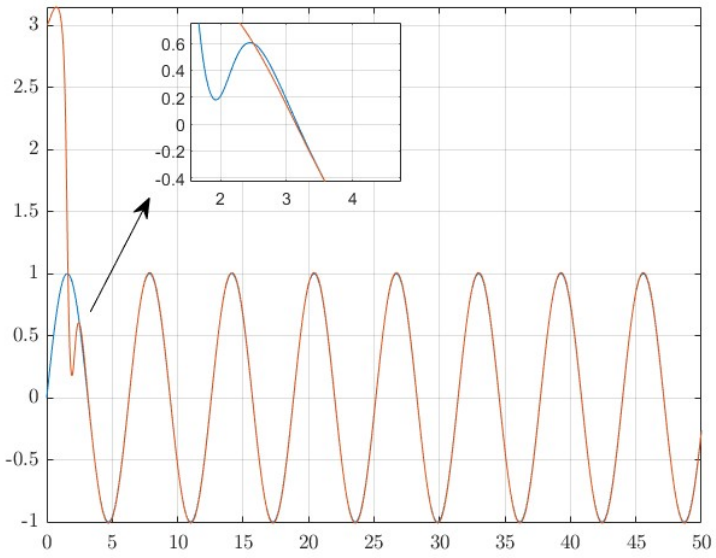
The input signal is a sinusoidal wave, $x_{des} = 1 * \sin(t)$. The results of each sliding surface are reported in the following table,

Table 3.1: Comparison of Control Techniques

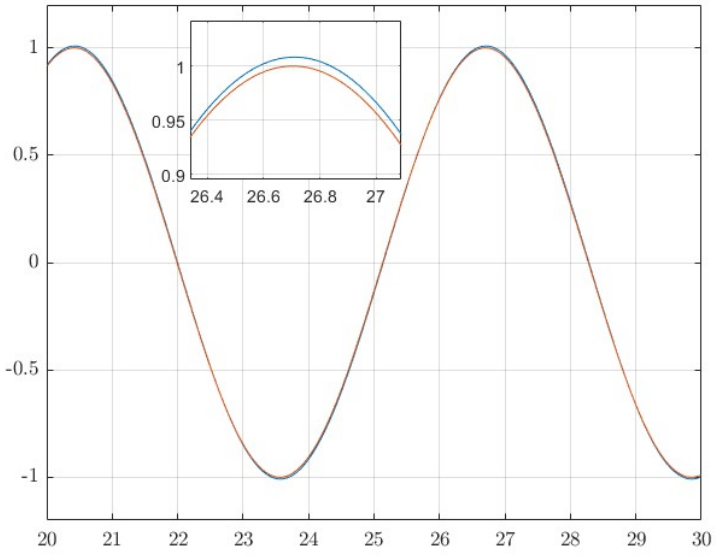
	IAE	ISV	$t_s[s]$
SMC	6.957	$6.27*10^4$	3.2
TSMC	3.925	$5.75*10^4$	2
FTSMC	3.759	$1.74*10^5$	2
NTSMC	3.787	$1.74 * 10^5$	24.2
NFTSMC	4.397	338	2.3

Also, it is possible to see the response of the system with each control law in the following figures, where the orange signal is the actual and the blue one is the desired state to be tracked.

Both table and the figures show that all the advance sliding surfaces are better than the classical ones. All the settling times are of the same order, even though the NFTSMC has slighter worse results error-wise but the control signal is one or two orders lower than the other advanced sliding surfaces, which is the requirement needed to be physically implemented.

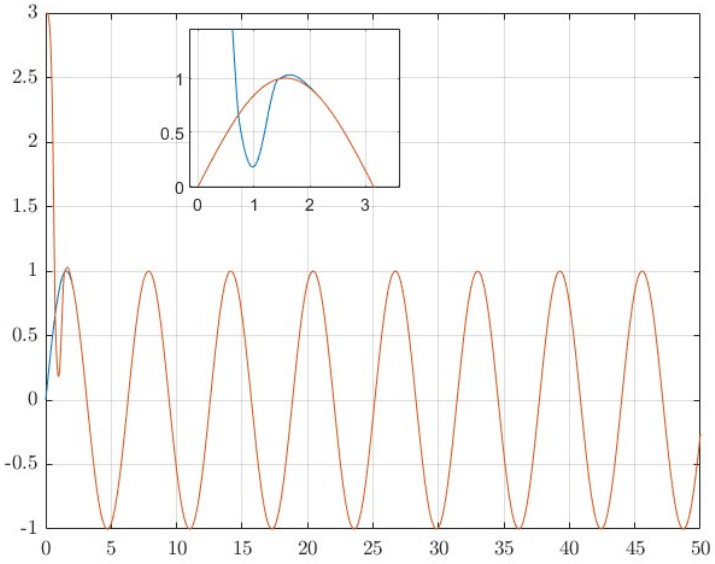


(a) Response of SMC, $k = 10$.

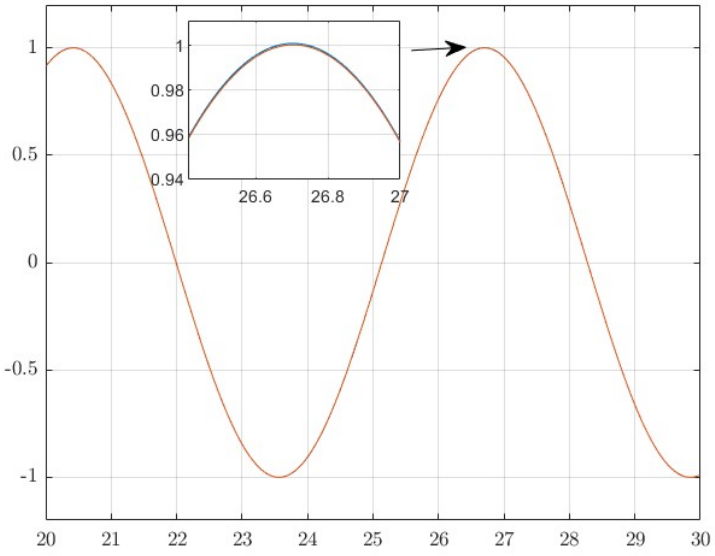


(b) Zoom in the range 20-30s.

Figure 3.2: Tracking performance of Classical Sliding Mode Control.

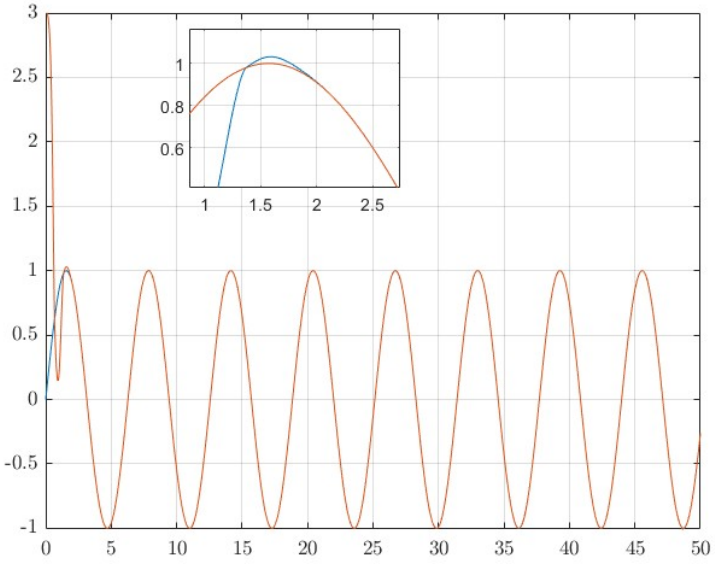


(a) Response of TSMC, $\beta = 10, q = 3, p = 5$.

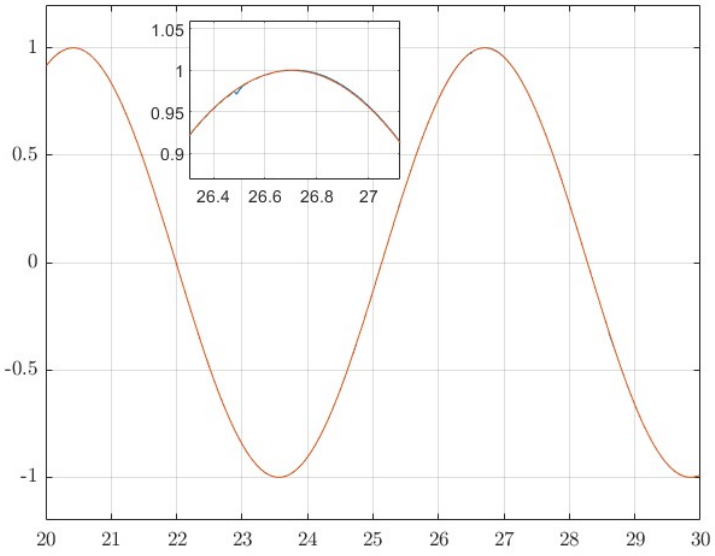


(b) Zoom in the range 20-30s.

Figure 3.3: Tracking performance of Terminal Sliding Mode Control.

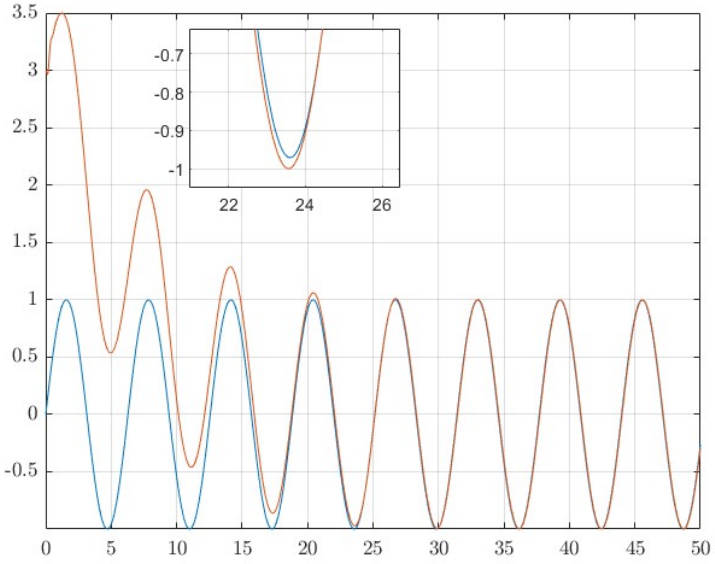


(a) Response of FTSMC, $\alpha = 0.1, \beta = 10$.

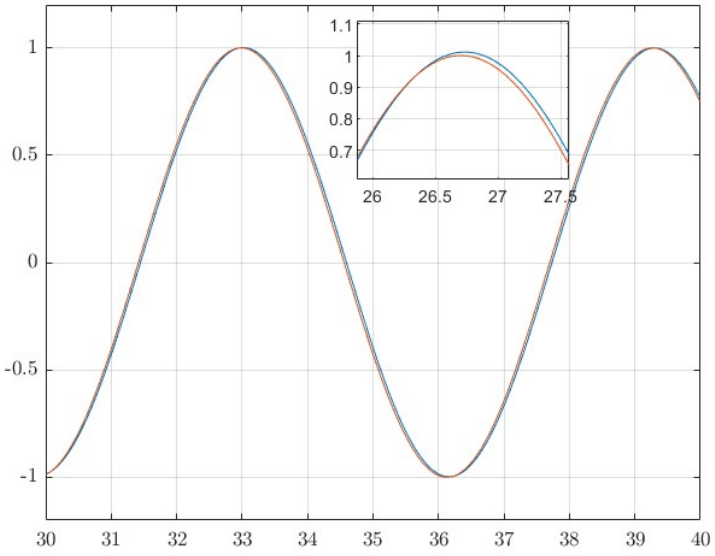


(b) Zoom in the range 20-30s.

Figure 3.4: Tracking performance of Fast Terminal Sliding Mode Control.

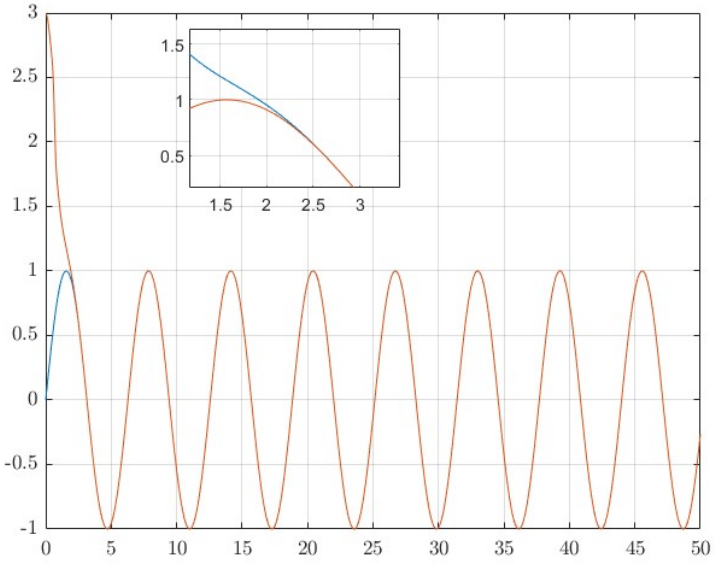


(a) Response of NTSMC, $\beta = 10, q = 7, p = 9$.

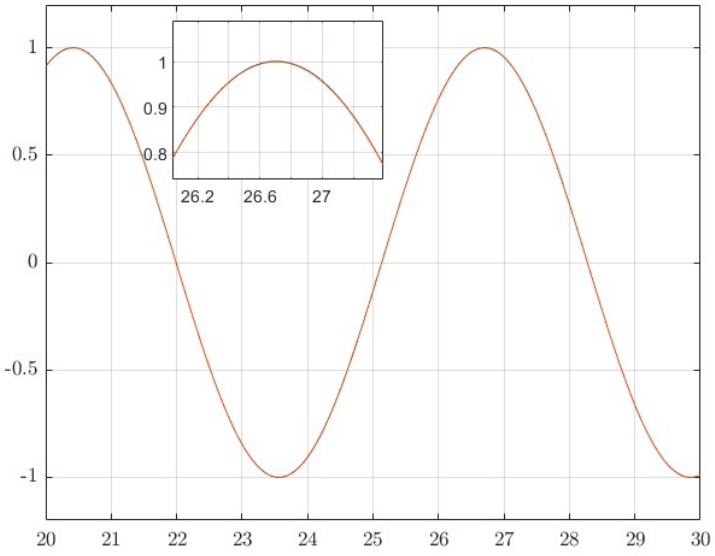


(b) Zoom in the range 30-40s.

Figure 3.5: Tracking performance of Non-Singular Terminal Sliding Mode Control.



(a) Response of NFTSMC, $k_1 = k_2 = 10, \alpha = 0.6, \beta = 0.95$.



(b) Zoom in the range 20-30s.

Figure 3.6: Tracking performance of Non-Singular Fast Terminal Sliding Mode Control.

Chapter 4

Brain Emotional Based Controller

4.1 Introduction

The main issue of the classic controller is that they require full knowledge of functions governing the dynamics of the system to be controlled. These functions are used to perform *feedback linearization*, i.e. mapping directly the states with the control law allowing to use of simpler control techniques. In real-world applications, the parameters do not always coincide with the ones used to design the controller and thus this can affect the tracking performance and stability of the system. The estimation of the dynamic functions is not an easy task and it is not always possible to do it offline, since some of the parameters might change during its application, like the mass of a car due to fuel consumption, for example.

Guaranteeing robustness, high tracking performance e low complexity of the algorithm is necessary for real-world purposes and applications, and much research has been done to provide solid theoretical background and practical usage.

In this regard, the last decades have been prolific for developing adaptive controls, which can adapt themselves according to the changing surrounding environment. One noteworthy adaptive control approach is the BELBIC presented in this thesis.

The objective of the BELBIC is to identify and compensate for model differences between the theoretical and actual assumptions. As already said, this system has lower complexity compared to other types yet high performance for estimating state-dependent functions. Inspired by the Limbic System of the mammalian brain, the simplified model containing only *Amygdala, OFC, Thalamus and Sensory Cortex* can provide the functions describing the dynamic and preparatory for the feedback linearization.

The sensory signal by means of a sliding surface, provides information on the state of the system to the main subsystem responsible for the adaptation, Amygdala and Orbitofrontal Cortex. These areas are modeled as Multi-layer with one hidden layer neural network and activation functions defined as Radial Basis Functions. The single hidden layer guarantees a $O(n)$ complexity. In the emotion-based neural network, the Amygdala is responsible for the so-defined excitatory network whereas the OFC is for the inhibitory one.

The input-output identification comes from comparing the outputs of the two subsystems, and it is based on learning the optimal weights linking the layers in the network. The learning process relies on the back-propagation algorithm and the updating law is chosen to be consistent with the biological behavior of the Amygdala and OFC and proven to guarantee closed-loop stability by means of Lyapunov stability theory. citare e aggiungere immagine

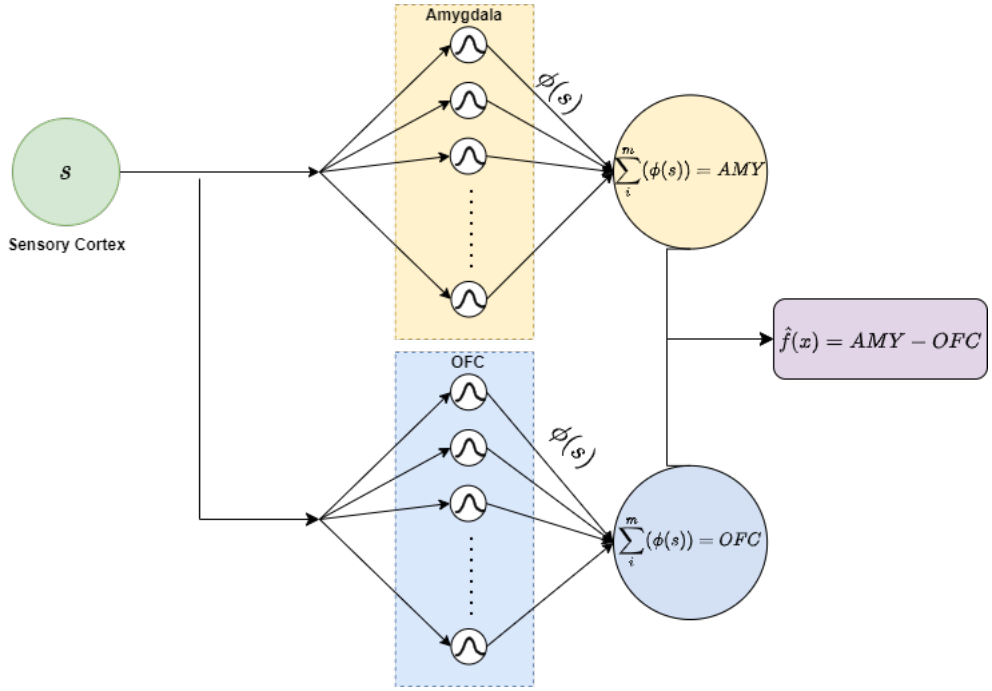


Figure 4.1: BELBIC scheme.

Consider an *affine*,γ-order SISO system

$$x^{(\gamma)} = f(x) + g(x)u + d(x,t)$$

The control law can be defined as

$$u = \frac{1}{g(x)}\left(f(x) - r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2\dot{e}\right)$$

, which requires the knowledge of $f(x)$ and $g(x)$.

What happens when the parameters change through time or the dynamic model is way easier than the real one?

A robust estimator is required to identify the functions of the dynamic to provide stabilization and tracking performance.

The general idea is depicted in Figure 4.2 and Figure 4.1. In the area of the *Sensory Cortex*, the sliding surface is fed by the error between the desired states and the actual ones, thus this information flows into the *Amygdala* and *Orbitofrontal Cortex*. The sliding surface enters each neuron in both subsystems which are characterized by *Radial-Basis*

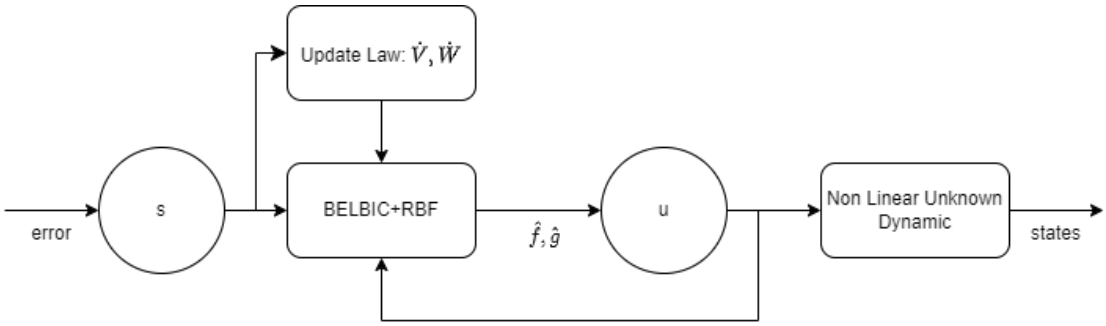


Figure 4.2: General control scheme with BELBIC.

activation function, $\phi(s)$, and the results of each subsystem is a weighted sum of the activation functions. The neurons of the Amygdala are indicated with the letter V, while the ones of the OFC are with the letter W. Within the Amygdala model, the *Thalamus* area is taken into account with an additional neuron with weight V_{th}.

The learning process, i.e. the update of the weights $\dot{V}$,$\dot{W}$ is based on the backpropagation algorithm and relies on the tracking errors and thus on the sliding surface.

The output of the BELBIC system, the estimate of the dynamic functions is provided, $\hat{f}(x)$,$\hat{g}(x)$, and the control law in eq.4.5 can be used to control a non-linear unknown γ-order system.

The information on the tracking error flows in the system through sliding surfaces, either the classic or the more advanced ones. In a γ-order system, the states are $x = \begin{bmatrix} x & \dot{x} & \cdots & x^{(\gamma-1)} \end{bmatrix}$, thus the classic sliding surface is defined as

$$s = e^{\gamma-1} + k_{\gamma-2}e^{\gamma-2} + \cdots + ke$$

, where $e = x - x_{des}$ is the tracking error.

Each neuron of the Amygdala and OFC receives this signal in its activation function, ϕ_i, which is Radial-Basis and it is activated according to it. Considering a neural network with m neurons,

$$\phi_i(s) = e^{\frac{\|s_j - c_i\|}{\sigma_i^2}} \qquad \forall\, i = 1, \ldots, m.\ and\ j = 1, \ldots, n.$$

, where n is the number of states to control.

The output of the Amygdala is a weighted combination of all the activation functions,

$$AMY = \sum_i^m V_i \phi_i(s) + V_{th}\Phi \tag{4.1}$$

, where $\Phi = max(\phi_1, \ldots, \phi_m)$ and it is used to model the Thalamus contribution in the Limbic system.

In the same fashion, the output of the Orbitofrontal Cortex is defined as,

$$OFC = \sum_i^m W_i \phi_i(s) \tag{4.2}$$

The estimate of the dynamic functions can be formed by the following equations,

$$\hat{f}(x) = AMY_f - OFC_f = \sum_i^m V_{i_f}\phi_{i_f}(s) + V_{th_f}\Phi_f - \sum_i^m W_{i_f}\phi_{i_f}(s) \tag{4.3}$$

$$\hat{g}(x) = AMY_g - OFC_g = \sum_i^m V_{i_g}\phi_{i_g}(s) + V_{th_g}\Phi_g - \sum_i^m W_{i_g}\phi_{i_g}(s) \tag{4.4}$$

It is now possible to define the control law

$$u = \frac{1}{\hat{g}(x)}\left(\hat{f}(x) - r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2\dot{e}\right) \tag{4.5}$$

The latter equation is a combination of the BELBIC system with the sliding mode control technique. It is noteworthy the absence of the additional v term, typical of the sliding mode control. The two techniques help each other to achieve tracking performance; the BELBIC system estimates the dynamic and provides the feedback linearization required to use the sliding mode control.

The input-output identification of the dynamic can be done with several techniques, for example, Neural Adaptive. The novelty in the use of the BELBIC depends upon the learning process of the weights in the neural networks, which is inspired by the Limbic system and thus the adaptation laws try to emulate its behavior, especially the contribution of the Amygdala. Indeed, the equation governing the BELBIC without the Amygdala is similar to the Neural Adaptive and can be extracted from Lyapunov stability analysis. Consider a first-order system described by a simple dynamic

$$\dot{x} = f(x) + u(t) \tag{4.6}$$

The weights of the neural networks are initialized to a random number and then adjusted by means of an optimization algorithm toward an optimal value which should drive to zero the estimation error, $\varepsilon = f(x) - \hat{f}(x)$.

These conditions can be described as follows

$$[V_f^*, W_f^*] = \underset{[V_f, W_f]}{argmin}\ \left(f(x) - \hat{f}(x)\right)$$

Under the conditions that the optimal values V_f^*, W_f^* lead to $\varepsilon = 0$ and that the function $f(x)$ can be exactly described by a combination of Radial-Basis functions, $f(x) = W^{*T}\phi(s)$, W^* and $\phi(s)$ with coherent dimensions, one can find the adaption laws based on the Lyapunov theory.

Consider a candidate Lyapunov function satisfying

$$V = \frac{1}{2}s^2 + \frac{1}{2}\tilde{W}^T\alpha^{-1}\tilde{W} \tag{4.7}$$

, where $s = x - x_{des}$ is the classic sliding surface and $\tilde{W} = W^* - W$ is the weight estimations errors. The subscript f is omitted since there is only one function to be estimated.

The derivative of the Lyapunov function,

$$\dot{V} = s\dot{s} + \alpha^{-1}\tilde{W}\dot{W}$$
$$\dot{V} = s(\dot{x} - \dot{x_d}) + \alpha^{-1}\tilde{W}\dot{W}$$

, which is required to be strictly negative to guarantee asymptotic stability. Substituting eq.4.6 in the latter equation, it is possible to evaluate the control input and also the learning law of the Orbitofrontal Cortex, $\dot{W}$.

$$\dot{V} = s(f(x) + u(t) - \dot{x_d}) + \alpha^{-1}\tilde{W}\dot{W}$$
$$\dot{V} = s(W^{*T}\phi(s) + u(t) - \dot{x_d}) + \alpha^{-1}\tilde{W}\dot{W}$$

The first term can be forced to negative by choosing the following control signal

$$u(t) = -W^T\phi(s) + \dot{x}_{des} - ks \tag{4.8}$$

, and substituting it in the derivative of the Lyapunov candidate,

$$\dot{V} = s(f(x) + u(t) - \dot{x_d}) + \alpha^{-1}\tilde{W}\dot{W}$$
$$\dot{V} = s(\tilde{W}\phi(s) - ks) + \alpha^{-1}\tilde{W}\dot{W}$$
$$\dot{V} = \tilde{W}s\phi(s) - ks^2 + \alpha^{-1}\tilde{W}\dot{W}$$

Finally, the learning law, or adaptation law, can be defined as

$$\dot{W} = -\alpha s\phi(s) \tag{4.9}$$

, α is the learning rate of the Orbitofrontal Cortex.

Regarding the Amygdala, the learning law has the same fashion but it must take into account its nature to be able to only learn.

$$\dot{V} = \beta \max(0, s)\phi(s) \tag{4.10}$$

, β is the learning rate of the Amygdala.

4.2 Comparison BELBIC vs Neural Adaptive

Consider a simple first-order system defined by the following dynamic

$$\dot{x} = f(x) + u(t) + d(t)$$

, where $f(x) = sin(x)$ and the control law is the same as eq.4.5.

The desired state is a pulse signal with a period of 20 seconds, a duty cycle of 50, and an amplitude equal to 1.

The BELBIC system is set with a number of neurons equal to $m = 50$ both for the Amygdala and OFC; the parameters of the Radial-Basis activation functions are $\sigma = 1$

and c is a random variable m dimension vector with mean value equal to 1. The sliding surface has a gain $k = 5$.

In order to validate the potentiality of the BELBIC system and thus of the Amygdala, a comparison has been made with a Neural Adaptive control. Results are reported in the next table, showing that the BELBIC provides better tracking performances, lower than 30% compared to the Neural Adaptive error and requiring almost the same magnitude in the control signal. The better performance is due to the presence of the Amygdala and its ability to have a memory of the past. Whenever the same scenario is present, this system is able to react faster and so the convergence time is smaller. This phenomenon is visible in Figure 4.3, where it is possible to notice that the overshoots in the tracking by the BELBIC are not symmetric and the upper peaks are smaller than the lower ones. This is not the case in the Neural Adaptive, where the overshoots are symmetric and equal to the lower peaks of the BELBIC.

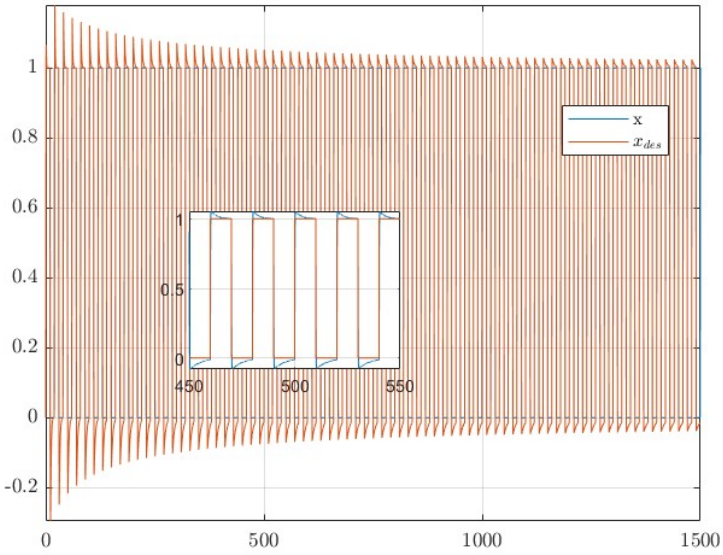
Furthermore, at the beginning of the simulation, the Neural Adaptive presents a worse tracking error, with peaks that are higher than 2.

The same results are obtained even an external time-dependent disturbance is applied to the system, which is a combination of three high-frequency sinusoidal signals. The BELBIC is not able to learn time-dependent signals but still, it tries to adapt itself to reduce the error and can better perform with respect to the Neural Adaptive.

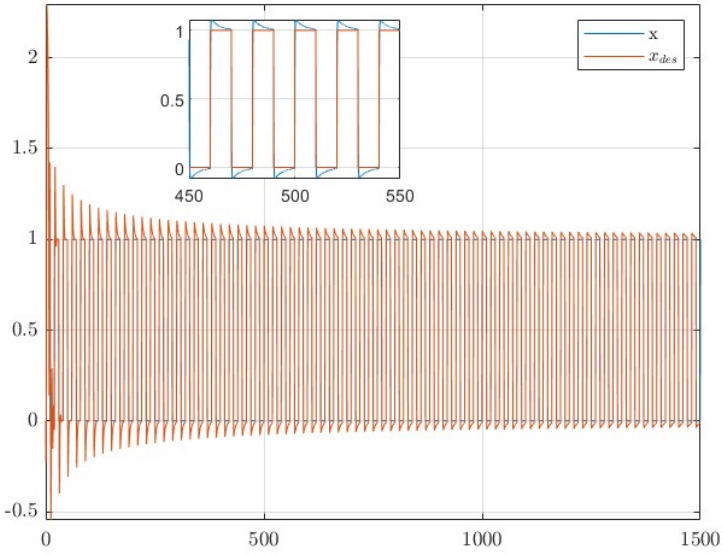
To increase the performance it is possible to consider an additional term taking into account the external disturbance, for example, $v = -k \tanh s$.

Table 4.1: Comparison of BELBIC and Neural Adaptive

Disturbance Off		
	IAE	ISV
BELBIC	44.36	1054
Neural Adaptive	65.70	972.9
Disturbance On		
	IAE	ISV
BELBIC	71.23	2371
Neural Adaptive	97.13	2282

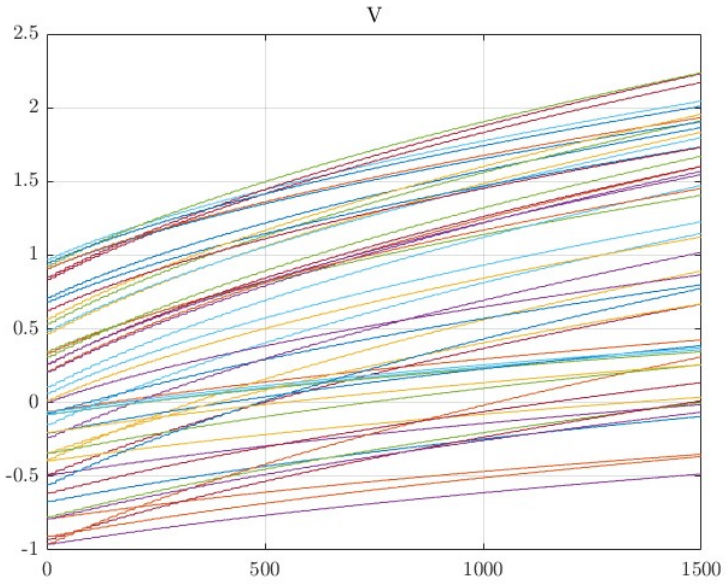


(a)

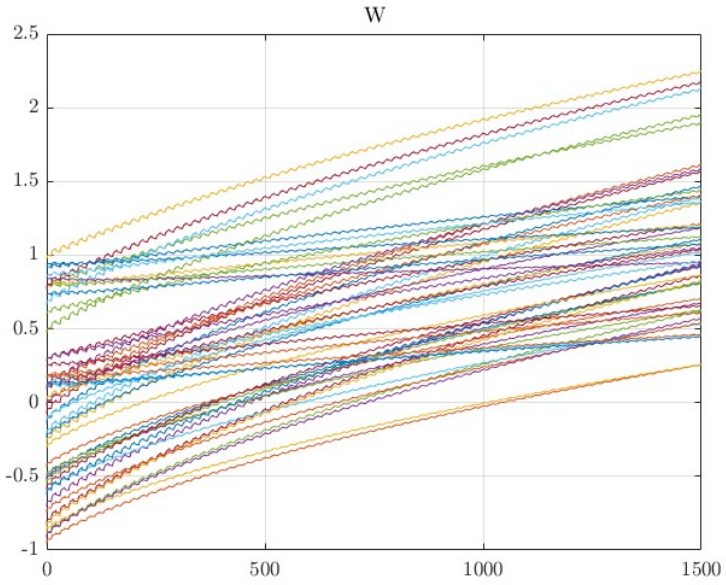


(b)

Figure 4.3: Tracking performance with no disturbance, BELBIC (a) and Neural Adaptive (b).



(a)

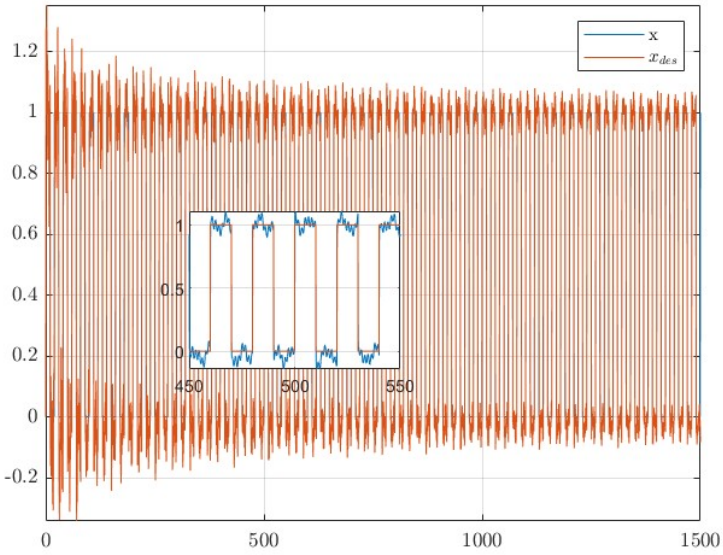


(b)

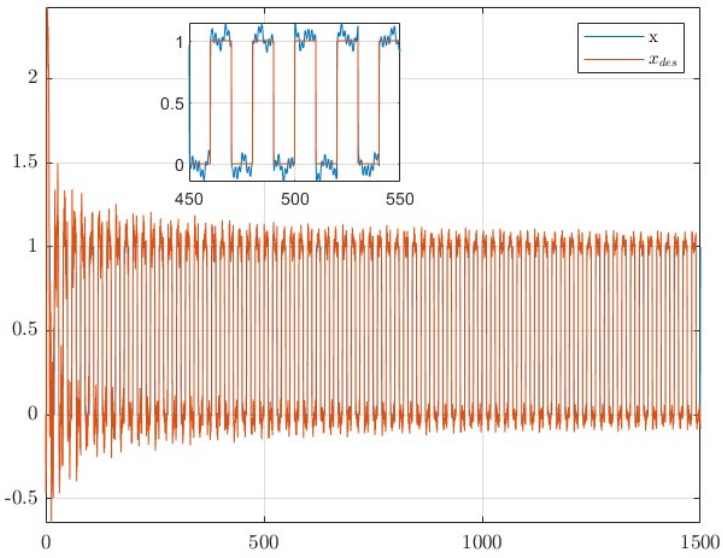
Figure 4.4: Learning weights of Amygdala (a) and Orbitofrontal Cortex (b).

4.3 Control of a mobile robot via BELBIC

In recent years, mobile robots have become widely used in industries thanks to their ability to walk in complex environments and their well-known structure. The simplest



(a)



(b)

Figure 4.5: Tracking performance with disturbance, BELBIC (a) and Neural Adaptive (b).

type of mobile consists of a body and two wheels used to walk in the $x - y\,plane$. The variables defining the state of the robot are the linear velocity v and the steering angle θ, the yaw angle about the $z - axis$.

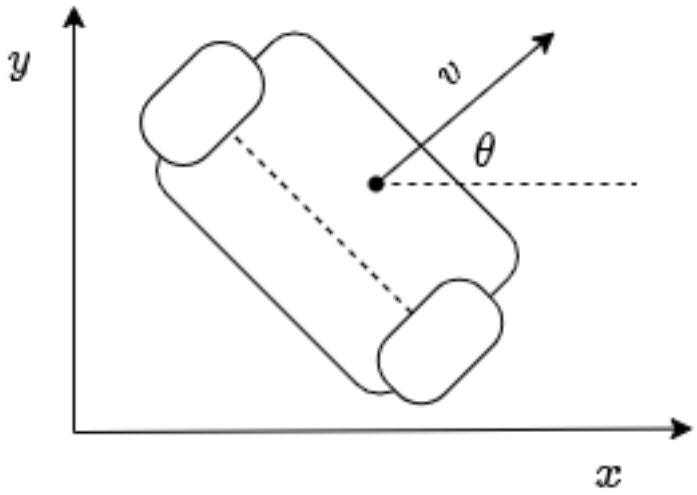


Figure 4.6: Representation of mobile robot.

According to Figure 4.6, the *Kinematics* of the mobile robot is defined by the following relations, which describe how its x and y positions change through the time

$$\dot{x} = v * cos\theta \tag{4.11}$$

$$\dot{y} = v * \sin\theta \tag{4.12}$$

In this thesis, the mobile robot is the virtual leader of the fleet of 4 drones and so it is important to control its position in space, leaving free the angle θ that it is controlled indirectly. For this purpose, one can implement the control scheme depicted in Figure 4.7 to control the virtual leader. Since it is not possible to directly control the kinematic of the system, a *Force Generator* should be used to compute the linear force F and the torque τ for the yaw angle, which are unknown. The following equation links the unknown forces to the virtual control signals, a combination of sliding mode control and BELBIC, by means of two matrices A and B.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = B + A \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{4.13}$$

,

$$B = \begin{bmatrix} -vw\sin\theta - lw^2 cos\theta \\ vw\cos\theta - lw^2\sin\theta \end{bmatrix} \tag{4.14}$$

$$A = \begin{bmatrix} \cos\theta & -l\sin\theta \\ \sin\theta & l\cos\theta \end{bmatrix} \tag{4.15}$$

The parameter l is used to avoid a singularity in the system and represent an offset of the body's reference system from the body's center of mass. The virtual control signals

are defined as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} u_{sx} + u_{BELBIC,x} \\ u_{sy} + u_{BELBIC,y} \end{bmatrix} \tag{4.16}$$

Both the terms are functions of the position tracking error and thus of the sliding surface.; the first term u_s is the control signal of a typical sliding mode control and so it also depends on the type of sliding surface. The second term arises from the BELBIC equations described by eq. 4.5 .

It is now possible to derive the required forces to track the desired position of the mobile robot by solving for F and τ, which are consequently integrated to provide the necessary v and θ to the kinematic block.

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = A^{-1} \left(\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} - B \right) \tag{4.17}$$

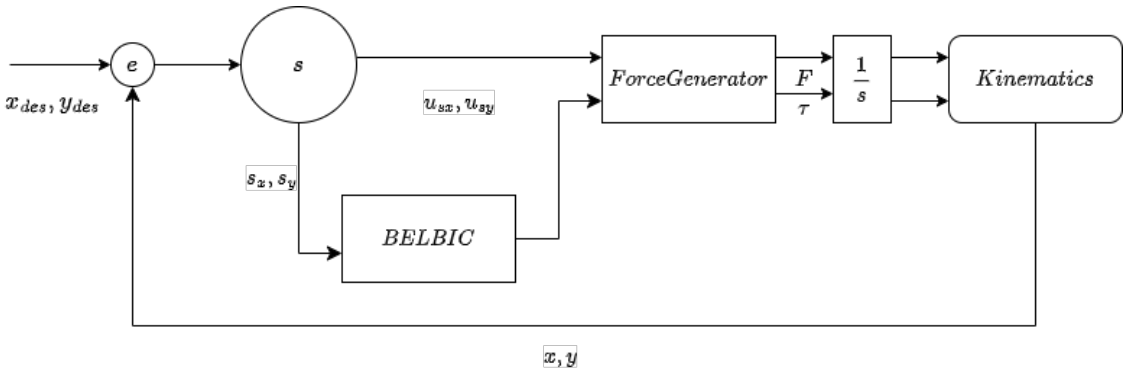


Figure 4.7: Control scheme of a mobile robot.

4.4 Control of a drone via BELBIC

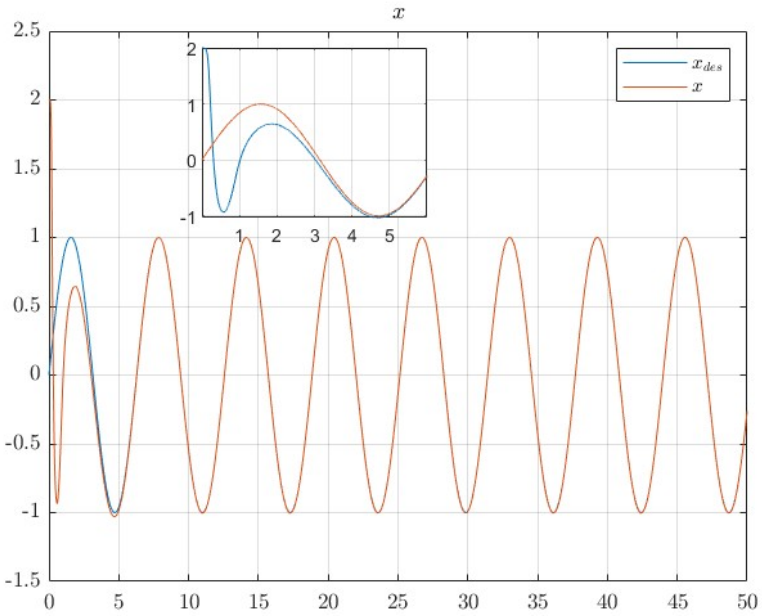
In the last years, *Unmanned Aerial Vehicle, UAV,* has reached more success thanks to their agility, maneuverability, and abilty to fly and discover places that otherwise could be really difficult to reach.

These abilities are used in different fields of the industry and with several goals.

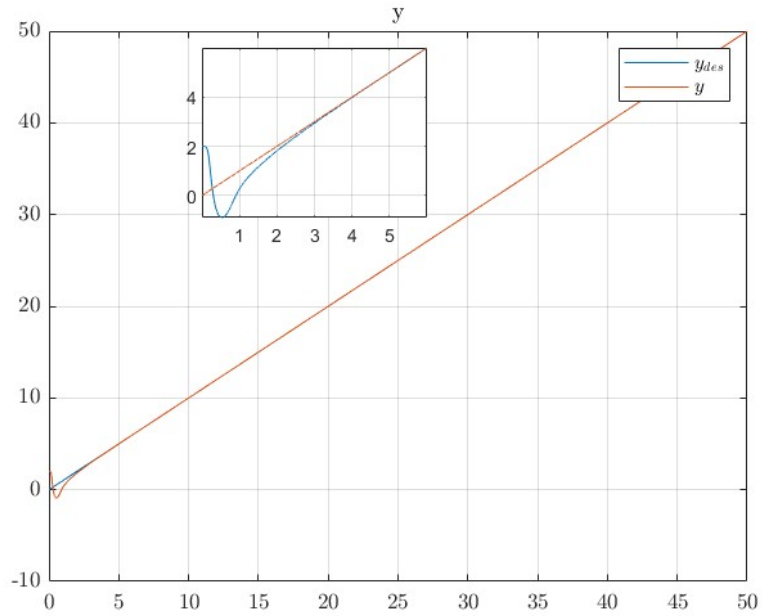
One of the main advantages of using drones can be found in environmental monitoring, such as biodiversity and wildlife, *Remote Sensing*, search and rescue operations, territorial security, borders, and the fight against drug traffickers.

The drone is part of the *Rotorcraft* family, which also includes helicopter and autogyro, and it is classified according to the number of motors: a *quadcopter* has 4 motors, a *hexacopter* has 6 motors and an *octocopter* has 8 motors.

Besides this classification, most of the drones have the same set of components. The commonly used embedded sensor to measure the distance from a surface is the ultrasound, which throws a signal to the ground and estimates the distance by considering the time taken to sense it back.



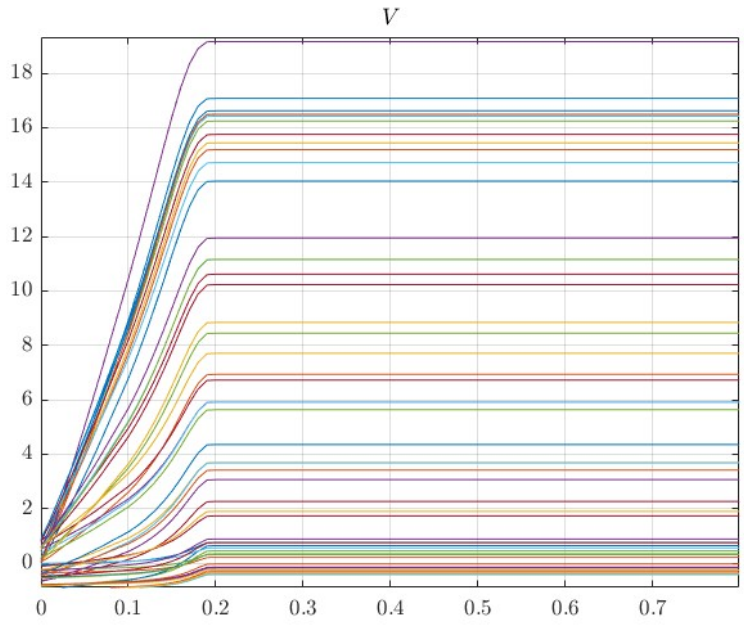
(a)



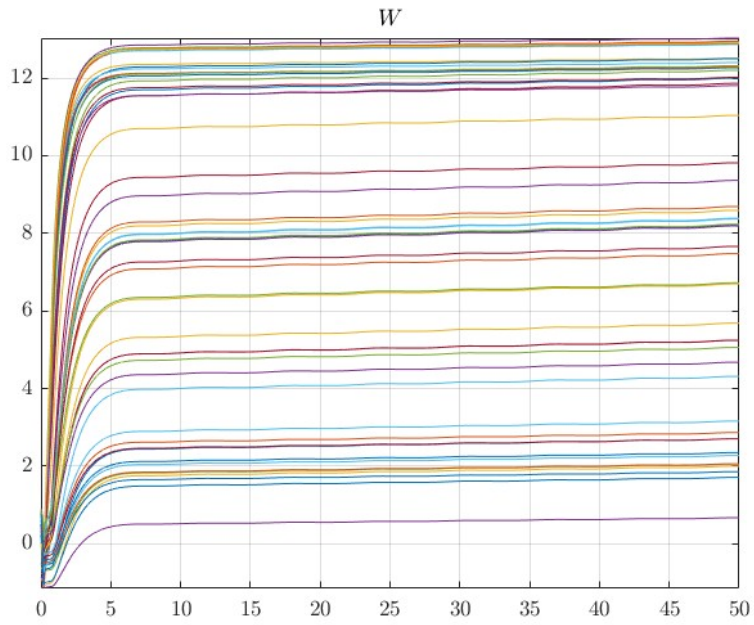
(b)

Figure 4.8: Control results of the mobile robot.

The horizontal displacement is computed by considering two consecutive frames of a



(a)



(b)

Figure 4.9: Weights of the BELBIC neurons: V,Amygdala (a) and W,OFC (b).

camera and computing the difference in the position of an object, while the accelerations

are computed using an IMU.

The altitude of the drone is then measured by a pressure sensor since it changes with the distance from the ground.

The drone considered in this thesis is a quadcopter and this means that is an underactuated system since it has 4 motors for 6 degrees of freedom. Consequently, translational and rotational motions are coupled: a quadcopter drone can not linearly move without affecting the angles.

4.5 Attitude and Position control of a quadcopter

The attitude and position of a drone are completely defined when all three positions and angles are known, $[x, y, z, \phi, \theta, \psi]$. They are described by computing the coordinate of its *roll, pitch, yaw* body's angles in an inertial orthonormal reference frame, defined by the origin O and a basis made of three versors (x, y, z). The body frame is the defined by $B(x_B, y_B, z_B)$ It is possible to find the coordinate of a reference frame in another reference

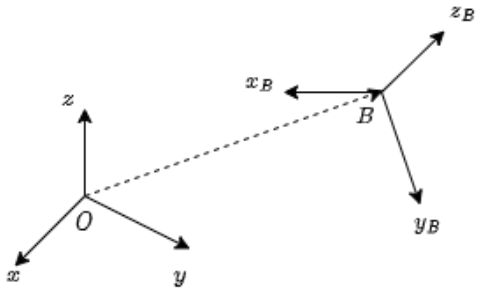


Figure 4.10: Rotation of a reference frame.

frame by means of a rotation matrix, usually indicated with the letter R_i^k, which indicated the rotation from the frame i to k.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_B^O \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{4.18}$$

, and R_B^O is the rotation matrix from the body to the inertia frame and it is computed by means of three simpler rotations, $R_B^O = R(x)R(y)R(z) = R(\phi)R(\theta)R(\psi)$.

$$R_B^O = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}$$

, where $c() = cos()$ and $s() = sin()$

The mathematical description of the dynamic of a drone is depicted in the following equations.

$$m_i \dot{v}_i = -m_i g + U_{T,i} + d_v(t) \tag{4.19}$$
$$J_i \dot{w}_i = -w_i^* J_i w_i + \Gamma_i + d_w(t) \tag{4.20}$$

, where J_i is the inertia matrix, m is the mass, U_T and Γ_i are the control coming from the actuators, and $d(t)$ is the external disturbances.

To control a drone, simply providing a desired linear position in space is not enough due to coupled motion between angles and position. A given desired position should correspond desired angles necessary to reach that specific point in the space. That means that the control scheme of a drone is made of two parts or *loops*: the *outer loop* takes care of the error position of the drone and so it provides the required thrust, U_T to achieve a zero tracking error. By means of geometric consideration, these forces are then used to compute the desired angles, and the control of these angles is made in the *inner loop*.

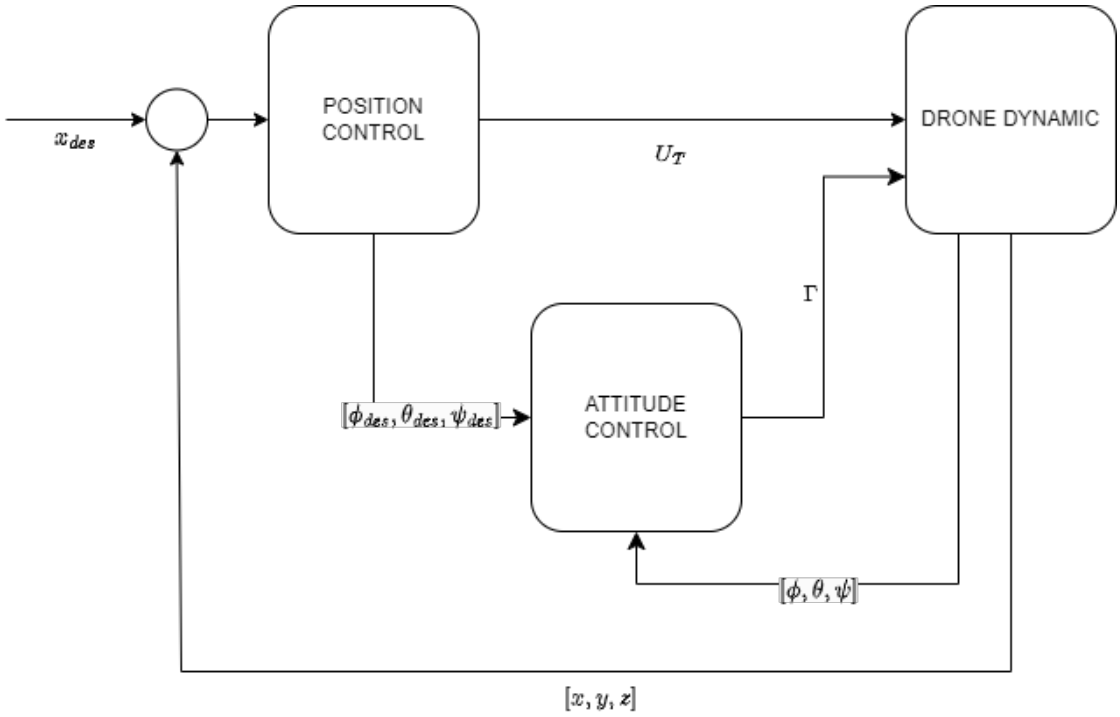


Figure 4.11: General Control scheme of a drone.

4.5.1 Position Control block via BELBIC

The dynamic of a drone can be written in the form

$$\dot{v} = f(x) + g(x)u(t) + d(t)$$
$$\dot{w} = h(x) + a(x)\tau(t) + d(t)$$

, and so the BELBIC system can be adopted to estimate the function of the system for feedback linearization.

The identification of the position functions relies on the information on the sliding surface. Given a desired position, the vector tracking error is defined as

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_{des} \\ y_{des} \\ z_{des} \end{bmatrix}$$

, and thus it is possible to define the classic sliding surface

$$\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} \dot{e_x} + k_x e_x \\ \dot{e_y} + k_y e_y \\ \dot{e_z} + k_z e_z \end{bmatrix}$$

Consequently, the identification can be done using eq. 4.3 and eq. 4.4, rewritten in the following

$$\hat{f}(x) = AMY_f - OFC_f = \sum_i^m V_{i_f}\phi_{i_f}(s) + V_{th_f}\Phi_f - \sum_i^m W_{i_f}\phi_{i_f}(s)$$
$$\hat{g}(x) = AMY_g - OFC_g = \sum_i^m V_{i_g}\phi_{i_g}(s) + V_{th_g}\Phi_g - \sum_i^m W_{i_g}\phi_{i_g}(s)$$

, and the control input defined as in eq. 4.5

$$u = mR_O^B \left(\frac{1}{\hat{g}(x)} \left(\hat{f}(x) - r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2 \dot{e} \right) \right)$$

, where R_O^B is the rotation matrix from the inertial frame to the body frame.

This signal corresponds to the required thrust to achieve the desired position and it is a row vector with 3 elements, one for each dimension in the space.

The final step of the outer loop is to provide the desired angle to the inner loop, and they can be computed according to geometric consideration,

$$\begin{bmatrix} \phi_{des} \\ \theta_{des} \\ \psi_{des} \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{u(1)}{u(3)+g}\right) \\ \arctan\left(\frac{-\cos\theta_{des} ut(2)}{u(3)+g}\right) \\ 0 \end{bmatrix}$$

, $\psi_{des} = 0$ because the drone should not spin about the $z - axes$.

4.5.2 Attitude Control block via BELBIC

After the desired angles are computed by the outer loop, the inner loop uses these references to calculate angle tracking error and so the sliding surface that will be provided to the BELBIC. The steps to be followed to estimate the functions for the angles dynamic are the same as previously shown.

$$\begin{bmatrix} e_\phi \\ e_\theta \\ e_\psi \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} - \begin{bmatrix} \phi_{des} \\ \theta_{des} \\ \psi_{des} \end{bmatrix}$$

$$\begin{bmatrix} s_\phi \\ s_\theta \\ s_\psi \end{bmatrix} = \begin{bmatrix} \dot{e_\phi} + k_\phi e_\phi \\ \dot{e_\theta} + k_\theta e_\theta \\ \dot{e_\psi} + k_\psi e_\psi \end{bmatrix}$$

$$\hat{h}(x) = AMY_h - OFC_h = \sum_i^m V_{i_h}\phi_{i_h}(s) + V_{th_h}\Phi_h - \sum_i^m W_{i_h}\phi_{i_h}(s)$$

$$\hat{a}(x) = AMY_a - OFC_a = \sum_i^m V_{i_a}\phi_{i_a}(s) + V_{th_a}\Phi_a - \sum_i^m W_{i_a}\phi_{i_a}(s)$$

$$u = J\left(\frac{1}{\hat{a}(x)}\left(\hat{h}(x) - r^{(\gamma)} - k_\gamma e^{(\gamma-1)} - \cdots - k_2\dot{e}\right)\right)$$

, where J is the inertia matrix.

The technique depicted so far took into account a classic sliding surface but obviously can be extended to all the advanced ones. In the simulation, for the position control, both NTSMC and SMC are used and compared, whereas, for the attitude, only the classic one is used.

The parameters of the sliding surface for the position used for the simulation in MATLAB/Simulink 2022b are:

- SMC, $k = diag([30{,}20, 50])$
- NTSMC, $k = diag([30, 20, 50]), q = 7, p = 13$

The parameters of the sliding surface for the attitude

- SMC, $k = diag([0.5, 0.1, 1])$

Simulation results are reported in the next tables and Figures.

Table 4.2: Position errors and control signals in a drone.

	SMC	NTSMC
e_x	2.914	2.005
e_y	36.89	22.28
e_z	3.561	1.894
u_x	48.93	39.09
u_y	10.32	31.73
u_z	8165	8220

Table 4.3: Attitude errors and control signals in a drone.

	SMC	NTSMC
e_x	4.994	2.983
e_y	6.445	3.917
e_z	0.4566	0.1635
u_x	4.458	2.351
u_y	1.737	1.325
u_z	4.501	5.46

4.6 Control of Heterogeneous MAS via BELBIC

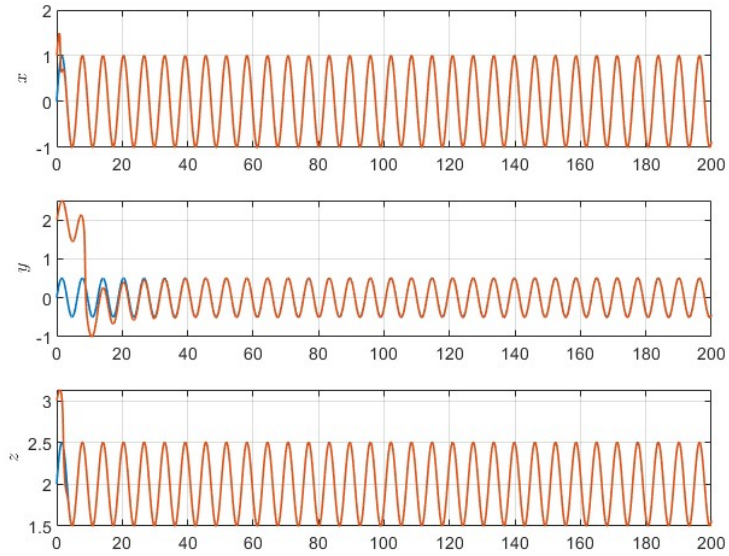
The main objective of this thesis is to use the BELBIC system to control a Heterogeneous Multi-Agent System composed of a fleet of 4 drones following a mobile robot as a "virtual leader".

For each agent and leader in the system, the BELBIC system is used to estimate the functions of the dynamic in the same manner as shown in the previous sections.

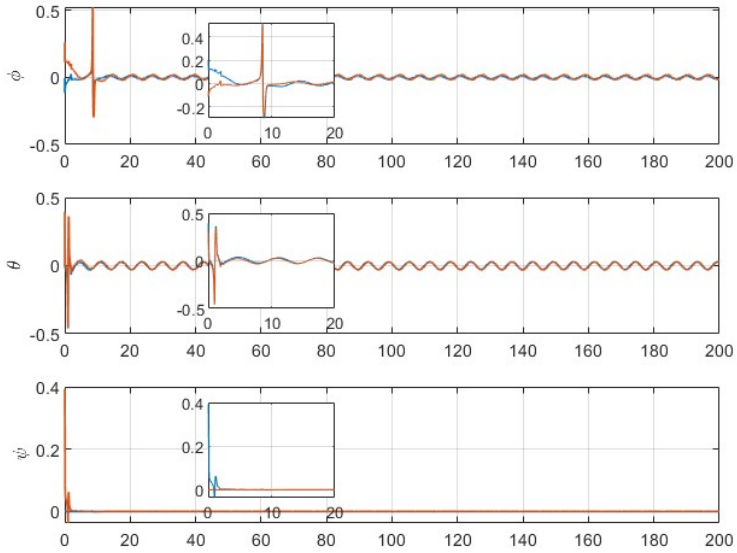
Figure 4.14 it is depicted the configuration used in this thesis. Agent 1 and 2 receive direct information from the leader, whereas drones 3 and 4 have a delay since they receive information from the previous two. Furthermore, agent 4 receive information both from agent 1 and 2 and thus can have redundant information, leading it to be the slowest one in the formation control. Indeed, agents 1 and 2 are supposed to be the fastest to reach the desired position coming from the leader, while agent 3 has some delay but is lower compared to agent 4.

To describe the overall state of the multi-agent system it is necessary to define the Laplacian matrix, L, which allows us to define the configuration of the graph. The communication protocol is completely defined considering the leader-agent matrix,B, which establishes whether an agent receives information from the mobile robot or not, and so the total matrix $H = L + B$. For the configuration in Figure 4.14, it is possible to use again the matrices defined in the first chapter of the thesis.

After computing these matrices, it is now possible to define the control signal that will manage the whole multi-agent system. Each i_th agent's position is defined by the



(a) Position of the drone.

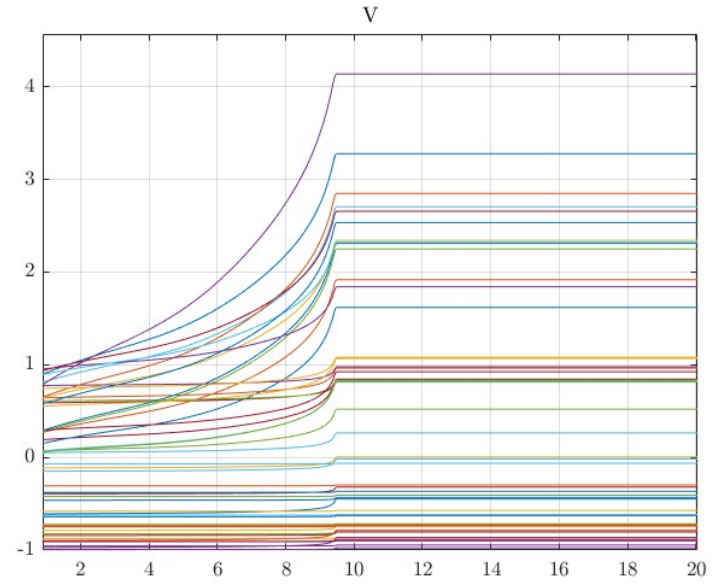


(b) Attitude of the drone.

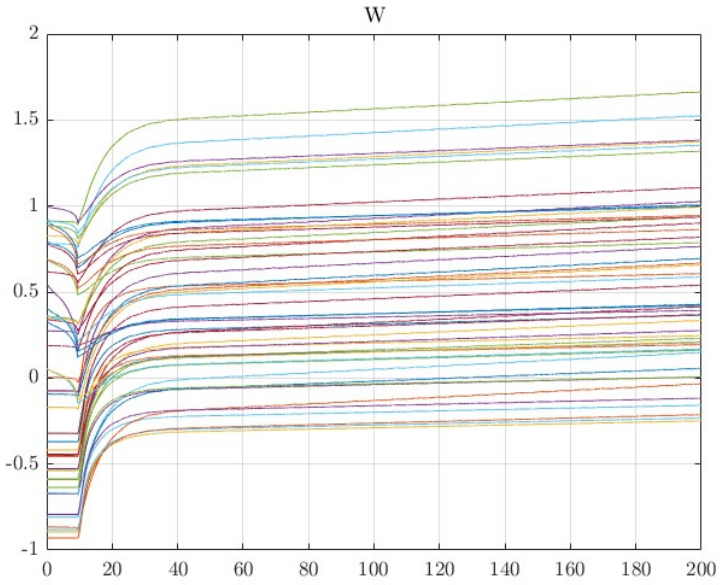
Figure 4.12: Attitude and position of the drone via BELBIC with NTSMC.

following vector

$$p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$



(a)



(b)

Figure 4.13: Weights of BELBIC neurons for the x position, Amygdala (a) and OFC (b).

, and the desired position to be tracked comes from the mobile robot

$$p_0 = \begin{bmatrix} x_{des} \\ y_{des} \\ z_{des} \end{bmatrix}$$

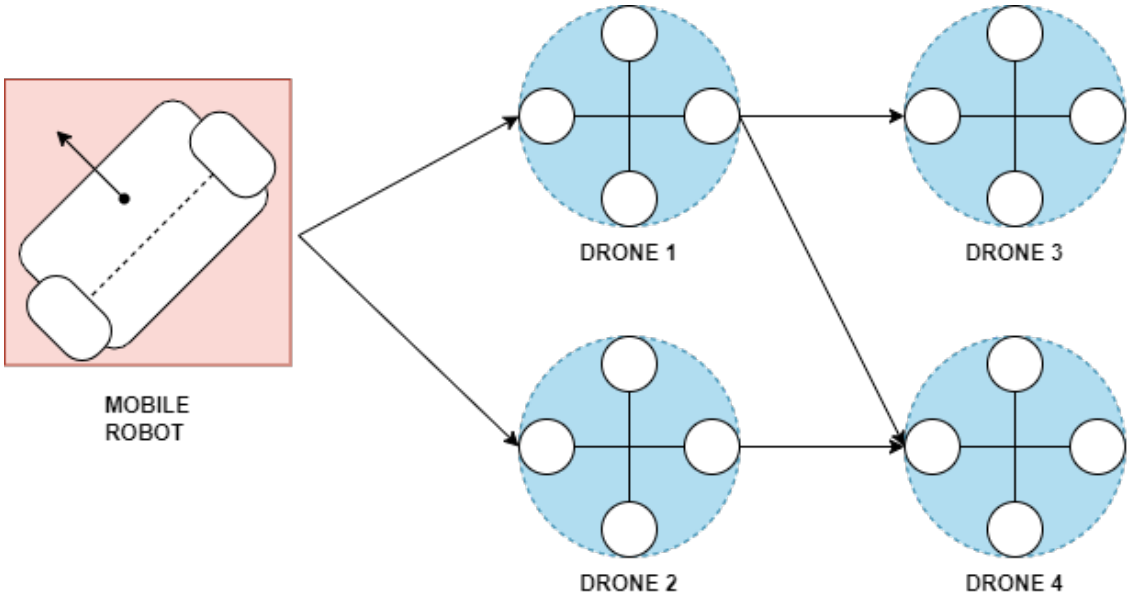


Figure 4.14: Etherogeneous Multi-Agent System.

.

The lumped tracking error is then defined as follows,

$$e_p = H \otimes I_3(p - 1_{n\times 1} \otimes p_0) \tag{4.21}$$

$$e_v = \dot{e_p} \tag{4.22}$$

, where $\otimes$ denotes the Kronecker product,I_n denotes an identity matrix $n \times n$, p is a vector with position in the space as element and thus has dimension $\Re^{3n\times 1}$ and n is the number of agents.

An example of how is possible to compute the control input for a consensus and flocking leader-agents is held for a classic sliding surface but it is valid for all the sliding surfaces presented in this thesis.

Given the following sliding surface,

$$s = e_v + ke_p$$

, s is a $\Re^{3n\times 1}$ vector, and k is a diagonal matrix with the elements of the gains for the control. The control signal can be then defined as

$$u(t) = H^{-1} \otimes I_3(\hat{f}(x) - ke_v) \tag{4.23}$$

, and the signal coming from the BELBIC system is the same as previously described. In the simulation, the Non-Singular Fast Terminal Sliding Mode Control is used to feed the BELBIC system and has the following parameter:

- $\alpha = 0.6$; $\beta = \alpha$+0.2;

- k2 = 0.1*diag([20, 30, 20, 20, 30, 20, 20, 30, 20, 20, 30, 10]);
- k1 = 0.1*diag([30, 15, 50, 30, 15, 50, 30, 15, 50, 30, 15, 50]);

In this thesis, the drones form a square-shaped scheme where each drone occupies a vertex of the square, on a plane that is above 1 meter the virtual leader. In the formation control, the lumped error should be modified according to the desired configuration, taking into account the relative distance of each drone from the mobile robot leader,

$$e_p = H \otimes I_3(p - 1_{n\times 1} \otimes p_0 - p_e) \tag{4.24}$$

, where $p_e = [0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0]$ is a row vector of the relative distance.

Furthermore, after 30 seconds, an external impulsive force is applied to the first agent both in the z and y axis.

The results of the simulation are reported in the next figures

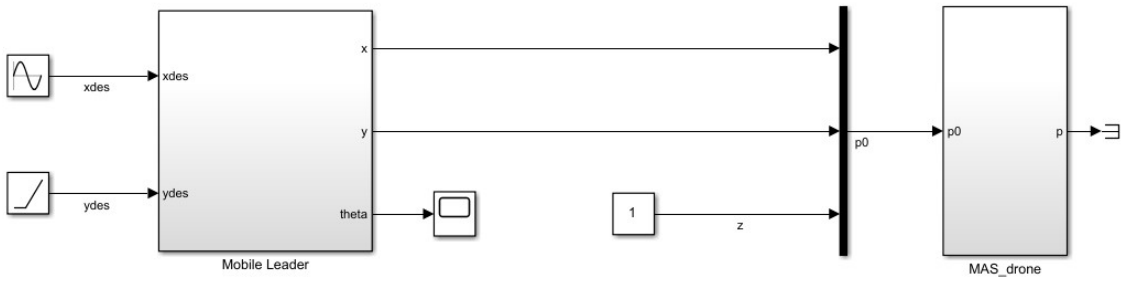


Figure 4.15: Simulink scheme of MAS.

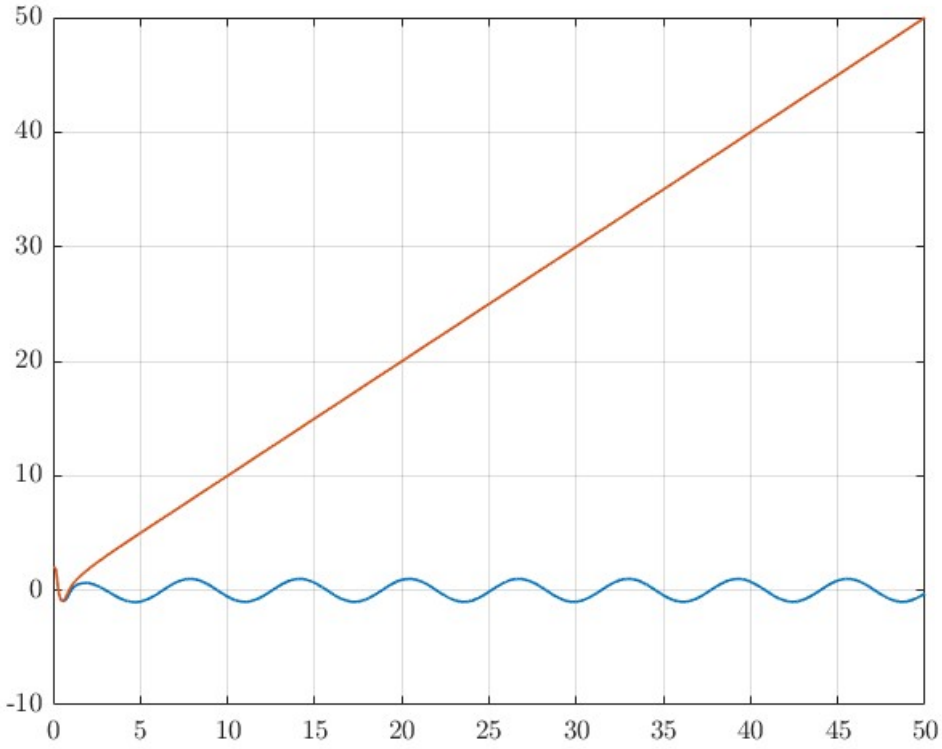


Figure 4.16: Position of the mobile leader.

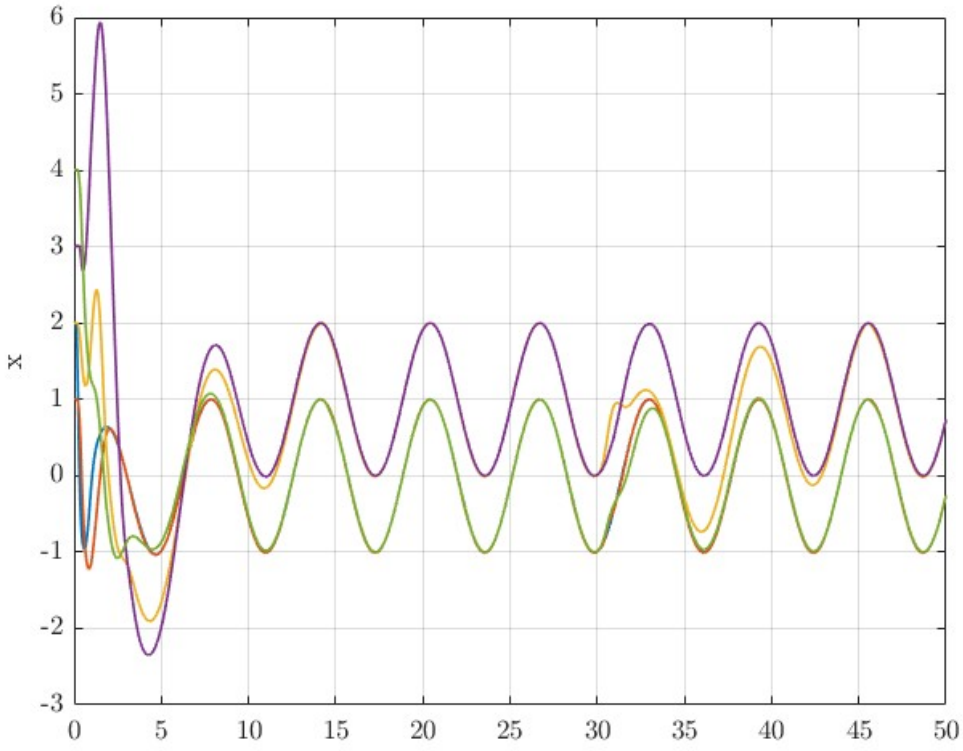


Figure 4.17: x position of the agents.

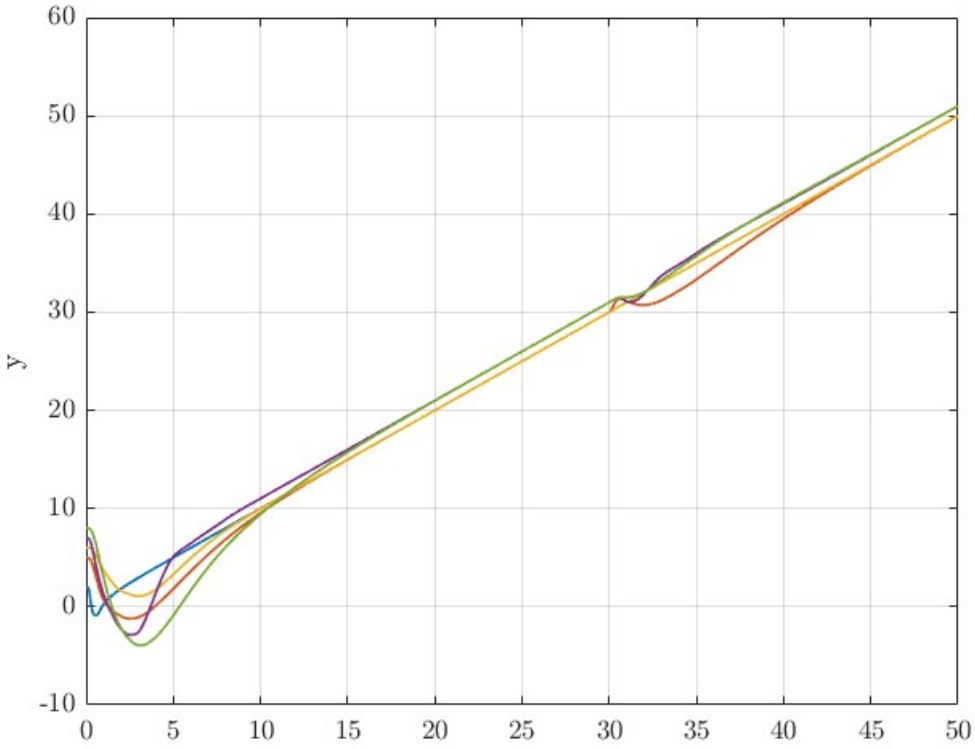


Figure 4.18: y position of the agents.

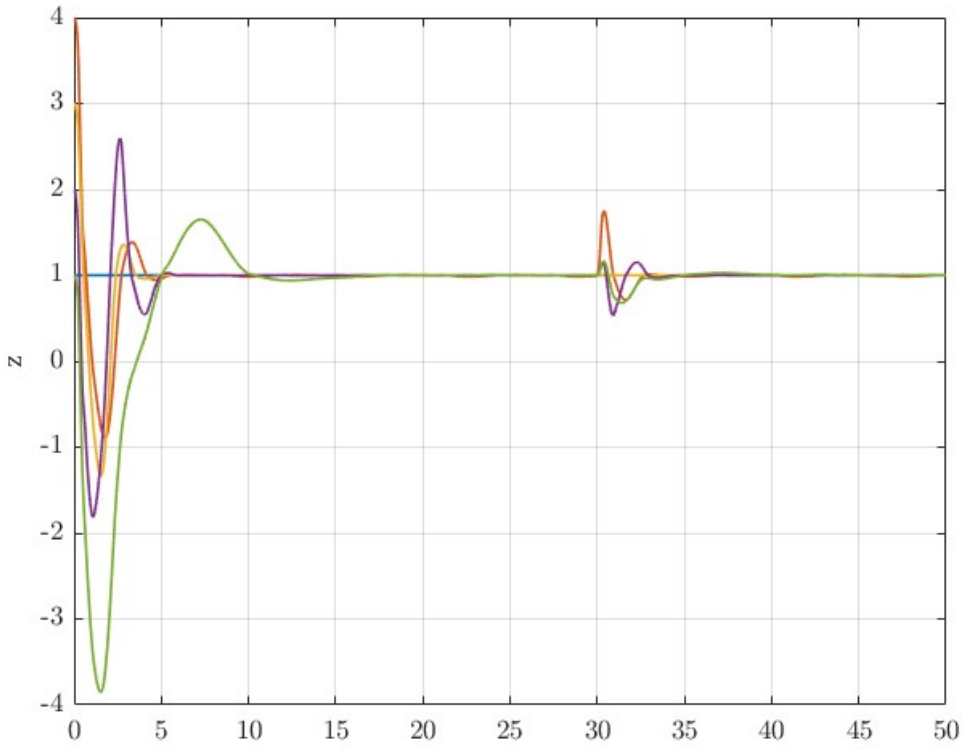


Figure 4.19: z position of the agents.

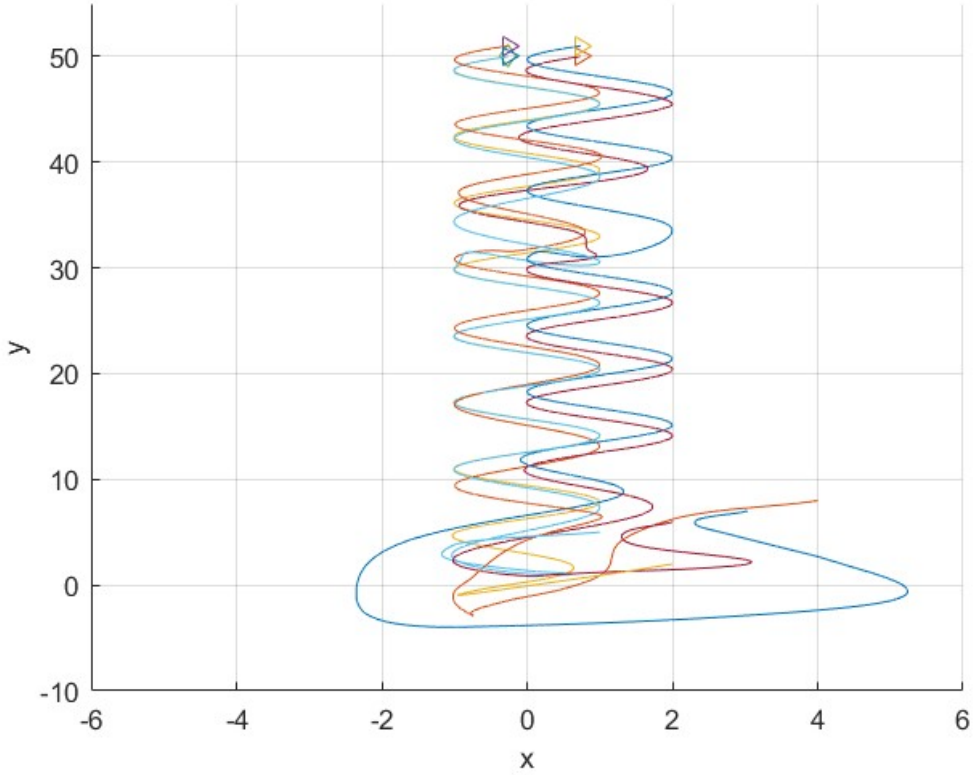
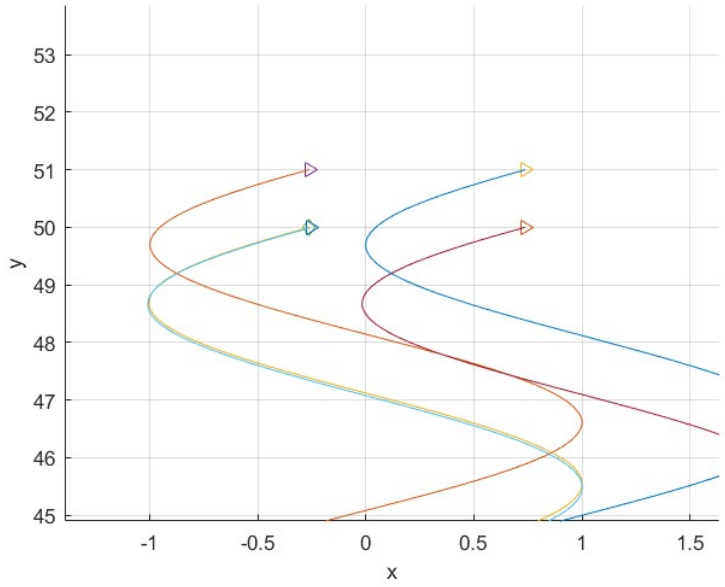
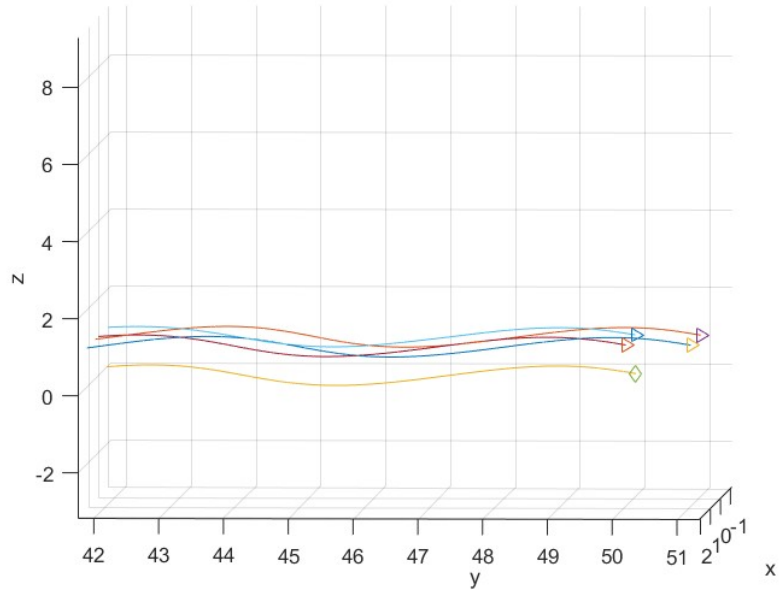


Figure 4.20: Formation of the Heterogeneous MAS.



(a)



(b)

Figure 4.21: Zoom of the MAS configuration.

Chapter 5

Conclusion

In conclusion, this thesis focused on the development of a BELBIC (Brain Emotional Sliding Mode Control) controller for the control of a fleet of four drones with a mobile robot as a virtual leader in a heterogeneous multi-agent system. The main objective was to implement a robust and efficient controller that would allow the drones to follow and maintain a stable formation around the mobile robot.

The obtained results demonstrate how the BELBIC system is indeed more performant compared to a Neural Adaptive system, which was also used for the identification of the dynamics' functions, providing robustness to the system in the presence of uncertainty in the dynamic parameters and external disturbances.

The BELBIC system is based on the limbic system located in the brain and consists of the Thalamus, Sensory Cortex, Amygdala, and OFC. It is thanks to the latter two areas that excellent performance is possible. Specifically, the adaptability laws of these two systems and the way they are interconnected allow the system to track past experiences and thus obtain better responses over time.

The area defined by the Sensory Cortex provides the Amygdala and OFC with information about the tracking error of the system through a sliding surface. For this reason, an analysis of different types of sliding mode control has been considered, analyzed, and compared. The results show that by adding an attractive terminal term to the equation of the classical sliding surface, the system can reach desired states in finite time. Furthermore, it is also possible to further modify the sliding surface to make the convergence time smaller while avoiding singularities.

Specifically, the Non-Singular Fast Terminal Sliding Surface has been used for the control of the multi-agent system, providing excellent results, with the drones hovering one meter above the mobile robot and forming a square formation with each drone occupying a vertex.

The system is also robust against simulated external disturbances, with a force acting on the first drone in the z and y axes. The drones are able to reform the square configuration and continue following the virtual leader.

Additionally, it can be observed how the communication protocol influences the response of each drone. In particular, agent 4 is consistently slower in following the virtual leader as it receives redundant information from drones 1 and 2. The latter, instead, are

more responsive to the commands received from the mobile robot as they directly receive information about the position to follow.

In future work related to this field, it could be interesting to consider a more elaborate and detailed model of the limbic system. In particular, a model of the Thalamus that allows for optimal utilization of its capabilities, such as signal filtering, could be highly relevant.

Bibliography

[1] Orval Mowrer. *Learning theory and behavior.* John Wiley & Sons Inc, 1960.

[2] Edmund T. Rolls. A theory of emotion, and its application to understanding the neural basis of emotion. *Cognition & Emotion*, 4:161–190, 1990.

[3] Jan Morén and Christian Balkenius. A computational model of emotional learning in the amygdala. 01 2000.

[4] Caro Lucas, Danial Shahmirzadi, and Nima Schei. Introducing belbic: Brain emotional learning based intelligent controller. *Intelligent Automation Soft Computing*, 10:11–21, 01 2004.

[5] Manik Chandra and Reza Langari. Analytical study of a control algorithm based on emotional processing. volume 2006, page 6 pp., 07 2006.

[6] S. Jafarzadeh, M.R. Motlagh, Mojtaba Barkhordari Yazdi, and R. Mirheidari. A new lyapunov based algorithm for tuning belbic controllers for a group of linear systems. pages 593 – 595, 07 2008.

[7] Maziar Ahmad Sharbafi, Caro Lucas, and Roozbeh Daneshvar. Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40:630 – 638, 12 2010.

[8] Changwon Kim and Reza Langari. Target tracking control of a mobile robot using a brain limbic system based control strategy. pages 5059–5064, 12 2009.

[9] Sophie Klecker, Bassem Hichri, and Peter Plapper. Robust belbic-extension for trajectory tracking control. *Journal of Mechanics Engineering and Automation*, 7, 02 2017.

[10] Mohammad Jafari and Hao Xu. Biologically-inspired intelligent flocking control for networked multi-uas with uncertain network imperfections. *Drones*, 2:33, 09 2018.

[11] Mohammad Jafari, Hao Xu, and Luis Rodolfo Garcia Carrillo. Brain emotional learning-based intelligent controller for flocking of multi-agent systems. pages 1996–2001, 05 2017.

[12] Ignacio Rubio Scola, Luis Rodolfo Garcia Carrillo, Joao Hespanha, and R. Lozano. Performance-guaranteed consensus control inspired by the mammalian limbic system for a class of nonlinear multi-agents. 07 2020.

[13] Tor Lattimore, Marcus Hutter, and Peter Sunehag. The sample-complexity of general reinforcement learning. *30th International Conference on Machine Learning, ICML 2013*, 08 2013.

[14] Robert M Sanner and Jean-Jacques E Slotine. Gaussian networks for direct adaptive control. In *1991 American control conference*, pages 2153–2159. IEEE, 1991.

[15] Bong Seok Park, Ji-Wook Kwon, and Hongkeun Kim. Neural network-based output feedback control for reference tracking of underactuated surface vessels. *Automatica*, 77:353–359, 2017.

[16] Ignacio Rubio Scola, Luis Rodolfo Garcia Carrillo, and JoÃo P. Hespanha. Limbic system-inspired performance-guaranteed control for nonlinear multi-agent systems with uncertainties. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2021.

[17] Yuri Shtessel, C. Edwards, Leonid Fridman, and Arie Levant. *Y. Shtessel, C. Edwards , L. Fridman, A. Levant. Sliding Mode Control and Observation, Series: Control Engineering, Birkhauser:Basel, 2014, ISBN: 978-0-81764-8923.* 01 2016.

[18] Irmgard Flugge-Lotz. *Discontinuous Automatic Control.* 01 2015.

[19] S.K. Korovin and Vadim Utkin. Using sliding modes in static optimization and nonlinear programming. *Automatica*, 10:525–532, 09 1974.

[20] R.A. DeCarlo, S.H. Zak, and G.P. Matthews. Variable structure control of nonlinear multivariable systems: a tutorial. *Proceedings of the IEEE*, 76(3):212–232, 1988.

[21] Rozaimi Ghazali, Mohd Rahmat, and Abd Hashim. Performance comparison between sliding mode control with pid sliding surface and pid controller for an electro-hydraulic positioning system. *International Journal on Advanced Science, Engineering and Information Technology*, 1, 01 2011.

[22] Vadim Utkin and Hoon Lee. Chattering problem in sliding mode control systems. *IFAC Proceedings Volumes*, 39(5):1, 2006. 2nd IFAC Conference on Analysis and Design of Hybrid Systems.

[23] Murat Furat and Ilyas Eker. Experimental evaluation of sliding mode control techniques. *Ãukurova Univ J Fac Eng Archit*, 27:23–37, 01 2012.

[24] Balaji Parasumanna Gokulan and D. Srinivasan. *An Introduction to Multi-Agent Systems*, volume 310, pages 1–27. 07 2010.

[25] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8:345–383, 2000.

[26] Lynne E Parker. Heterogeneous multi-robot cooperation. Technical report, Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 1994.

[27] Jonatan Uziel Alvarez-Munoz, Juan Antonio Escareno, Jeremy Chevalier, Steven Daix, and Ouiddad Labanni-Igbida. Wind-tolerant event-based adaptive sliding-mode control for vtol rotorcrafts multiagent systems. *IEEE Transactions on Aerospace and Electronic Systems*, 59(2):1400–1410, 2023.

[28] J. U. Alvarez-Munoz, J. Chevalier, Jose J. Castillo-Zamora, and J. Escareno. Distributed event-based sliding-mode consensus control in dynamic formation for vtol-uavs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1364–1373, 2021.

[29] Housheng Su, Xiaofan Wang, and Zongli Lin. Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, 2009.

[30] Daniel Spielman. Spectral graph theory. *Combinatorial scientific computing*, 18:18, 2012.

[31] Frederic B. Fitch. Warren s. mcculloch and walter pitts. a logical calculus of the ideas immanent in nervous activity. bulletin of mathematical biophysics, vol. 5 (1943), pp.

115â133. *The Journal of Symbolic Logic*, 9(2):49â50, 1944.

[32] Hugo Yanez-Badillo, Ruben Tapia-Olvera, and Francisco Beltran-Carbajal. Adaptive neural motion control of a quadrotor uav. *Vehicles*, 2:468–490, 07 2020.

[33] Hugo Yanez-Badillo, Francisco Beltran-Carbajal, Ruben Tapia-Olvera, Antonio Favela-Contreras, Carlos Sotelo, and David Sotelo. Adaptive robust motion control of quadrotor systems using artificial neural networks and particle swarm optimization. 09 2021.

[34] Daniel Galvan-Perez, Hugo Yanez-Badillo, Francisco Beltran-Carbajal, Ivan Rivas, Antonio Favela-Contreras, and Ruben Tapia-Olvera. Neural adaptive robust motion-tracking control for robotic manipulator systems. *Actuators*, 11:255, 09 2022.

[35] Francisco Beltran-Carbajal, Hugo Yanez-Badillo, Ruben Tapia-Olvera, Julio Caro, Carlos Sotelo, and David Sotelo. Neural network trajectory tracking control on electromagnetic suspension systems. *Mathematics*, 11:2272, 05 2023.

[36] M. Powell. Radial basis functions for multivariable interpolation. *Algorithms for Approximation of Functions and Data*, pages 143–167, 01 1987.

[37] David Broomhead and D Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 01 1988.

[38] B. Krose and Patrick van der Smagt. *An Introduction to Neural Networks*. 01 1996.

[39] E. T. Rolls. A theory of emotion and consciousness, and its application to understanding the neural basis of emotion. pages 1091–1106, 1995.

[40] J. E. LeDoux. In search of an emotional system in the brain: leaping from fear to emotion and consciousness. pages 1049–1061, 1995.

[41] Ali Sadeghieh, Jafar Roshanian, and Farid Najafi. Implementation of an intelligent adaptive controller for an electrohydraulic servo system based on a brain mechanism of emotional learning. *International Journal of Advanced Robotic Systems*, 9:1, 09 2012.

[42] Reshma Ravi and Mija S.J. Design of brain emotional learning based intelligent controller (belbic) for uncertain systems. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pages 1089–1093, 2014.

[43] Pankaj SHARMA and Vinod KUMAR. Design of novel belbic controlled semi-active suspension and comparative analysis with passive and pid controlled suspension. *Walailak Journal of Science and Technology (WJST)*, 18, 03 2021.

[44] Shirin Aghaee, Caro Lucas, and Khosrow Zadeh. Applying brain emotional learning based intelligent controller (belbic) to multiple-area power systems. *Asian Journal of Control*, 14, 11 2012.

[45] Jae-Won Kim and Chae-Youn Oh. Control of a rotary inverted pendulum system using brain emotional learning based intelligent controller. *Journal of manufacturing engineering technology*, 22, 10 2013.

[46] Xinghuo Yu, Yong Feng, and Zhihong Man. Terminal sliding mode control â an overview. *IEEE Open Journal of the Industrial Electronics Society*, 2:36–52, 2021.

[47] Venkat Shastri and Sandeep Gulati. Control of nonlinear systems using terminal sliding modes. *Journal of Dynamic Systems Measurement and Control*, 115, 02 1992.

[48] Xinghuo Yu and Zhihong Man. Model reference adaptive control systems with terminal sliding modes. *International Journal of Control - INT J CONTR*, 64:1165–1176, 08 1996.

[49] Zhihong Man and Xinghuo Yu. Terminal sliding mode control of mimo linear systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 44:1065 – 1070, 12 1997.

[50] Michail Zak. Terminal attractors for addressable memory in neural networks. *Physics Letters A*, 133(1):18–22, 1988.

[51] Michail Zak. Introduction to terminal dynamics. *Complex Systems*, 7(1):59, 1993.

[52] Xinghuo Yu and Zhihong Man. Zhihong, m.: Fast terminal sliding-mode control design for nonlinear dynamical systems. ieee trans. circuits syst. i fundam. theory appl. 49(2), 261-264. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49:261 – 264, 03 2002.

[53] Yong Feng, Xinghuo Yu, and Zhihong Man. Non-singular adaptive terminal sliding mode control of rigid manipulators. *Automatica*, 38:2159–2167, 12 2002.

[54] Mohamed Boukattaya, Neila Mezghani, and Tarak Damak. Adaptive nonsingular fast terminal sliding-mode control for the tracking problem of uncertain dynamical systems. *ISA Transactions*, 77, 04 2018.