

AN EMBODIED AND EXTENDED COGNITIVE DYNAMIC FIELD
THEORY (DFT) MODEL FOR PILOTING TASKS SUPPORTED WITH
FLIGHT RECORDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YASİN KAYGUSUZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

SEPTEMBER 2015

**AN EMBODIED AND EXTENDED COGNITIVE DYNAMIC FIELD
THEORY (DFT) MODEL FOR PILOTING TASKS SUPPORTED WITH
FLIGHT RECORDS**

Submitted by Yasin Kaygusuz in partial fulfilment of the requirements for the
Master of Science in Cognitive Science, Middle East Technical University by,

Prof. Dr. Nazife Baykal
Director, **Informatics Institute**

Prof. Dr. Cem Bozşahin
Head of Department, **Cognitive Science**

Y.Doç.Dr. Murat Perit Çakır
Supervisor, **Cognitive Science, METU**

Examining Committee Members:

Prof. Dr. Cem Bozşahin
Cognitive Science, METU

Y.Doç.Dr. Murat Perit Çakır
Cognitive Science, METU

Y.Doç.Dr. Cengiz Acartürk
Cognitive Science, METU

Doç. Dr. Tolga Esat Özkurt
Medical Informatics, METU

Y.Doç.Dr. Murat Ulubay
Management, Yıldırım Beyazıt University

Date: 08th September 2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Yasin Kaygusuz

Signature: _____

ABSTRACT

AN EMBODIED AND EXTENDED COGNITIVE DYNAMIC FIELD THEORY (DFT) MODEL FOR PILOTING TASKS SUPPORTED WITH FLIGHT RECORDS

Kaygusuz, Yasin
Department of Cognitive Science
Supervisor: Dr. Murat Perit Çakır

September 2015, 105 pages

The DFT modeling employs the dynamical systems approach to model the human motor and sensory stimulus unified with cognitive behavior. Since the DFT model includes the embodiment already embedded into the model, it can be easily used to generate task oriented robotic behavior similar to human activities including cognition. In this study the navigation trajectories of an aircraft with a pilot will be modeled using the DFT approach. The model will include the embodiment of the human mind into the body of the human and the aircraft. In an aircraft the sensory data required to maintain situational awareness is transferred to the pilot mostly via the displays. Additional tools may be aural and tactile systems. The DFT model in this study will include these cockpit tools as interacting layers in the DFT model following an extended mind thesis approach. Finally, in order to guide the development and verification of the trajectory equations deduced from the model within a simulated three-dimensional environment including waypoints and obstacles, real flight data records from various aircrafts will be used.

Keywords: Dynamic Field Theory, Pilot Cognition, Embodied Cognitive Model, Amari Equation, Pilot Cognitive Model.

ÖZ

UÇUŞ KAYITLARIYLA DESTEKLENMİŞ, DİNAMİK ALAN TEORİSİ (DAT) TEMELLİ, SOMUTLAŞMIŞ VE GENİŞLETİLMİŞ BİR BİLİŞSEL PİLOT MODELİ

Kaygusuz, Yasin
Bilişsel Bilimler Bölümü
Tez Yöneticisi: Dr. Murat Perit Çakır

Eylül 2015, 105 sayfa

Dinamik Alan Teorisi (DAT) insan bilişsel davranışını, motor ve algılayıcı uyaranlarla birlikte modellemede dinamik sistemler yaklaşımını kullanır. DAT doğası gereği somutlaştırmayı modelin içine gömülü olarak zaten içerdiğinden, bilişsel davranışı da içeren insan aktivitesini taklit eden, görev odaklı robot davranışı geliştirmede rahatlıkla kullanılabilir. Bu çalışmada pilot içeren bir uçağın seyrüsefer rotaları DAT yaklaşımı kullanarak modellenmiştir. Söz konusu model insan aklının ilgili işlevlerinin somutlaştırılmış bir modelini uçakla birlikte içerecektir. Bir uçakta durumsal farkındalık sağlamakta kullanılan algılayıcı çıktıları pilota genellikle göstergeler aracılığıyla iletilirler. İlave olarak sesli ya da dokunsal uyarıcı sistemler kullanılabilir. Bu çalışmadaki DAT modeli bu tip kokpit içi algılayıcı iletimlerini, genişletilmiş akıl tezi ile uyumlu bir şekilde etkileşim katmanları olarak içerecektir. Son olarak, modelin geliştirme ve doğrulamasına şık tutmak amacıyla, modelin üç boyutlu bir sanal ortamda engeller ve hedef durumlar arasında yaptığı seyrüseferin çıktısı olan izlerin gerçek uçuş testi sonuçları ile kıyaslanması verilecektir.

Anahtar Kelimeler: Dinamik Alan Teorisi, DAT, Pilot Bilişsel Modeli, Somutlaşmış Bilişsel Model, Amari Denklemi.

To My Family.

ACKNOWLEDGEMENT

The study provided in this thesis is supported by Turkish Aerospace Industries' (TAI) Technology Management Department. I express sincere appreciation to Turkish Aerospace Industries and all its Managers in duty, starting from Mr. Özcan Ertem, Mr. Akif Çetintaş, Mr. Eyüp Serdar Gökpınar and especially Mr. Özgür Altun for their encouraging and inspiring behaviour towards employees' academic studies. It may be acknowledged that supportive environment is a result of innovative change in the dynamics of TAI with the visions set by Mr. Muharrem Dörtkaşlı as the president and CEO. Of course I may thank to Dr. Murat Perit Çakır as my thesis supervisor and the only person who directed me towards dynamical systems in cognitive science for all his contribution, aid and patience. My teammates in TAI also provided me great encouragement. Finally, to my wife Deniz and daughters Sare and Elif Bade, I thank from my hearth for their patience in my long absences and continuous faith in me. Without them, I would not be such motivated to understand human cognitive behaviour.

This page is intentionally left blank.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ	v
ACKNOWLEDGEMENT	vii
TABLE OF CONTENTS.....	ix
LIST of FIGURES	xii
LIST of TABLES.....	xiv
LIST of CHARTS	xv
CHAPTERS	
1 INTRODUCTION	1
2 MODELLING AND UNDERSTANDING STUDIES OF PILOT COGNITION OR PILOTING TASKS	5
2.1 Control Models	5
2.1.1 Crossover model	6
2.1.2 Paper Pilot Model	8
2.1.3 Optimal Pilot Model	9
2.1.4 Generic Pilot Model.....	10
2.2 Studies on Cognitive Information Flow in the Cockpit	11
3 COGNITIVE MODELLING APPROACHES	15
3.1 Symbolic Architectures.....	15
3.1.1 ACT-R	16

3.1.2 SOAR (States, Operators And Reasoning).....	18
3.1.3 Embodiment Critics on Symbolic Architectures.....	19
3.1.4 Neural Network Based Pilot Models.....	20
3.2 Neural Models and Bio-inspired Models	21
3.2.1 Conductance Based Models	23
3.2.2 Spiking Neurons Modelling Approach.....	27
3.2.3 Networks of Neuron Models to Construct Intelligence.....	29
3.3 Rate Models.....	30
3.3.1 Dynamic Neural Field and Dynamical Field Theory	32
3.3.2 An Introduction to the Mathematics of Dynamical Systems.....	42
3.3.3 Temporal Characteristics of Cognitive Models.....	49
3.3.4 Task Hierarchy and Sequencing in DFT	51
3.3.5 Robotics Applications of DFT	53
3.4 Frameworks.....	55
3.4.1 Cosivina.....	55
3.4.2 CEDAR	56
4 STRUCTURE OF THE MODEL.....	59
4.1 The Aircraft Model.....	59
4.1.1 3 DoF Aircraft Model.....	59
4.2 Step 1: Constant Intention Pilot Model	62
4.3 Step 2: Improved Pilot Model	66
4.4 Step 3: Pilot Model with Altitude Effects on Hold Performance	67
4.5 Negative Deviations' Representation in The Spatiotemporal Axes	71
4.6 Network Elements	72
5 RESULTS, DISCUSSION AND CONCLUSION.....	75
5.1 Results	75
5.1.1 Understanding the Human Pilot Vertical Hold Performance	75
5.1.2 Model Outputs.....	80
5.1.3 Comparison Of Model and Pilot Performances.....	84

5.2 Discussion and Conclusion	85
5.2.1 Weakness of DFT Approach on Negative Axis Description	85
5.2.2 Weakness of CEDAR DFT Framework on Adaptive Model Parameters	86
5.2.3 Evaluation of DFT Approach for Pilot Modelling	88
5.2.4 Conclusion	92
5.2.5 Future Work	94
REFERENCES	95

LIST OF FIGURES

Figure 1. A representation of Crossover Model pilot-controlled aircraft element loop given with a slight modification from Blakelock (1991).	6
Figure 2. A schematic representation of the paper pilot.....	8
Figure 3. A schematic description of optimal pilot model is given.....	10
Figure 4. Inner and Outer loops of control are shown.....	11
Figure 5. A generic model of information flow between human agents and external artifacts.	14
Figure 6. Overview of ACT-R v5.0 (Andersen et al., 2004; Anderson, 2002).	17
Figure 7. The schematic representation of the main parts of a neuron.....	22
Figure 8. The representation of the neural electrical transmission direction and functions from a neuron to the next one.....	23
Figure 9. Time constant function amplitudes representation for gating variables of HH model.....	24
Figure 10. Gating variable representations for gating variables.....	25
Figure 11. Membrane current versus varying input voltage (mV) representation..	26
Figure 12. Representation of the HH neuron output spiking behaviour.....	26
Figure 13. The representation of the output for a Leaky Integrate-and-Fire (LIF) neuron model.....	29
Figure 14. The sigma node representation.	31
Figure 15. A dynamic system and states representation.....	34
Figure 16. The representation of an interaction kernel in form of a Gaussian with 0 amplitude global inhibition.	37
Figure 17. The representation of an interaction kernel with negative and constant inhibitory global behaviour.	38
Figure 18. A threshold function in form of a sigmoid.....	39
Figure 19. A Gaussian input (red solid line) with mean at point B, and amplitude A can be used as an input to Amari equation.	40

Figure 20. The representation of the formation of a kernel via the superposition of inhibitory and excitory interactions.	41
Figure 21. The representation of the steady state solution to Amari equation.	45
Figure 22. The steady state solution to Amari equation with hat type kernel when there is a Gaussian input to the system (Amplitude of the Gaussian input is 2.1, $\sigma=0.4$).	45
Figure 23. The representation of the multiple bump steady state of a field activity.	46
Figure 24. CEDAR Screenshot.	57
Figure 25. A Representation of a flying body neglecting the rotational forces for 3DoF motion model.	60
Figure 26. Acquisition/Perception Architecture.	63
Figure 27. Decision Architecture in Lateral Channel.	63
Figure 28. Motor Behaviour Summarized.	64
Figure 29. Complete Step 1 Model Architecture.	65
Figure 30. Obstacle Position Perception Architecture.	66
Figure 31. Detection Switch Mechanism Representation.	68
Figure 32. Overall Improved Pilot Model Representation.	69
Figure 33. Pilot Model with Altitude Effects on Performance.	70
Figure 34. The method of dividing a spatial variable into two fields for representation of different directions in an axis.	71
Figure 35. Dimension reduction function.	72
Figure 36. Histogram of absolute Values of Mean Errors.	78
Figure 37. Regression models: with full data set (up) and eliminated data set (down). ...	80
Figure 38. The relation between the DFT implementation of a spatiotemporal axis and original representation is shown	86
Figure 39. Altitude effects addition to the control output via a mechanical switch mechanism.	87
Figure 40. Biologically based adaptive field approach.	88

LIST OF TABLES

Table 1. Human Pilot Unfocused Vertical Hold Performance.	76
Table 2. Descriptive Statistics for Flight Test Results.	77
Table 3. Skewness and Kurtosis Statistics for Flight Test Results.....	77
Table 4. Pearson Correlations Between Target Altitude and Absolute Mean Deviation. .	78
Table 5. Eliminated test data set.....	79

LIST OF CHARTS

Chart 1. Mean deviation performance for model pilot. Normalized values represented..	81
Chart 2. Mean maximum deviation performance for both step 2 and 3 represented.	82
Chart 3. Model data sets' comparison above 5000 ft HAG.	83
Chart 4. Vertical position hold performance represented during a whole mission flight (10 minutes).	83
Chart 5. Human and model pilot data sets at different altitudes represented.	84

CHAPTER 1

1 INTRODUCTION

In present cognitive science literature, the classical problem of mind-body dualism still seems to await a resolution (Robbins and Aydede, 2009). The emergence of dynamical systems theory has opened up new avenues for researchers to explore new cognitive modelling frameworks that can potentially transcend this dualism. This relatively new approach may be seen as one of the important turning points in the history of cognitive science, which resembles the rise of the connectionist models against classical symbol-based architectures or models (Robbins and Aydede, 2009). The terms “situated cognition”, “embodied cognition” and “dynamical systems” are claimed to provide an understanding of the human cognition in unity with its environment, culture from which it is fed and other agents being natural or artificial. The dynamical systems and the embodiment approach to human cognition allows the design and implementation of biologically more plausible cognitive models without requiring the use of a symbolic layer. Such an approach allows the model to be in natural interaction with the sensory and motor stimuli and use the steady states of the dynamical system rather than symbols as states of the mind. By promising a solution to the “symbol grounding” problem, the embodied dynamical approach to cognitive modelling provides a conceptual framework for defining the relationship between the external world and the biological signals that underlie cognitive processing (Robbins and Aydede, 2009). The relation mentioned here is a transformation between the acquired external world and the embodied language of the biological body.

As far as the search for the concepts required to build an embodied description of such relations is concerned, Dynamic Field Theory offers a promising answer to many of the previously mentioned issues in cognitive science. The DFT is not alone in this endeavour as recent cognitive models that are based on networks of neurons (such as the one given in Hurzook, Trujillo and Eliasmith (2013)) have recently been proposed to address the same gap, whose similarities and differences will be further discussed in the literature review. Until today, most of the cognitive models have been in the form of symbolic architectures that process series of symbols specified in formal/computer languages.

Although such symbolic frameworks offer a mapping from human language into computer language, they may not necessarily capture all aspect of human cognition (e.g. graceful degradation, fault tolerance, parallel distributed processing) and useful mainly in the modelling of high-level, algorithmic aspects of cognition. In contrast, the Dynamic Field Theory and other theories that belong to the family of “embodied cognition” claim to provide a language for constructing a representational space for the brain in terms of neural spikes or electrical activity. Although the language mentioned here is not a set of meaningful symbols, it refers to the regularities of the processes through which the body biologically represents the external world inside the brain. Dynamic Field Theory as the subject of this thesis prefers to “represent” a spatiotemporal value belonging to the external world inside the body by transforming those values into cortical electrical activity (Schöner, 2007) and treats the steady state values of cortical fields as mental states. So it may be deduced that DFT provides a framework for cognitive modelling which does not require any algorithmic means of language and allows the researcher to construct a model with algorithms embodied inside the architecture.

One important advantage of using Dynamic Field Theory is that it allows the development of cognitive robotics applications not only in software but also at the implementation level encompassing sensor-processor interaction, opening up the possibility of truly cognitive robotics. In this thesis, a cognitive model of a human pilot is developed to observe if the Dynamical Field Theory may provide smart navigation control behaviour in 3D in aerial robotics. Although the goal of the study is geared towards a robotics application, the focus of the study is to build a cognitive model that renders the end product not just as an autopilot but a cognitive pilot “bot” that can achieve aircraft attitude control. In the literature, there are multiple robotic studies that can navigate successfully in 2D environments and such evidence is offers promise for successful 3D applications (refer to chapter 3.3.5 for robotic study examples). Therefore, in pursuit of a road map towards the ultimate goal of reaching cognitive aerial robotics, the current study presents a first step where the aim is to evaluate the pros and cons of Dynamic Field Theory based robotics in a 3D aviation context. The DFT based cognitive model of a pilot developed in this thesis is then evaluated empirically by comparing its attitude-control performance with the actual performance data obtained from human pilots. In that regard, the thesis also provides rare data on human attitude control performance for the purpose of model evaluation.

The current thesis study summarized in the paragraphs below belongs to the area known as computational cognitive modelling, with a focus towards modelling the navigational abilities and behaviour of a human pilot who performs a very sophisticated set of spatial perception and control tasks. The cognitive processes underlying piloting skills have been an attractive area of research in cognitive science, yet previous modelling studies of pilot cognition mostly use either cognitive architectures or control theoretic approaches, which do not share the same principles with the dynamical framework employed in this thesis study. Here the development, architecture and simulation of a Dynamical Field Theory (DFT) based pilot model and comparison of its outputs with real flight test results will be described. The main approach for the development of the given model is that it is an “embodied and situated” model using dynamical approach based upon Amari equation (Amari, 1977) and it is not an algorithmic model using representations processing, i.e. the developed model does not contain any “if/then statements or while/for loops” or similar algorithmic symbol manipulation primitives, and uses only field activations to simulate

embodied cognitive behaviour. In the paragraphs below, the content of each chapter is briefly summarized to provide an overview of this thesis study.

At this point, we should state that the DFT models given in this thesis are strictly limited to the attitude control and obstacle avoidance tasks and do not aim to capture any other piloting abilities or tasks. The model contains only a perception layer of aircraft attitudes in 3 axes, an awareness of altitude and the position of the obstacle if it exists. Other cues about the pilot and aircraft environment are completely neglected. The motor abilities of the pilot are assumed to be in direct interaction with the aircraft control surfaces. Finally, complex actions such as muscular tension for extremities angular control are left out of scope of this study.

The second chapter will provide an overview of existing modelling and simulation approaches in the context of aviation. Modern aviation uses various types of simulation technologies for various purposes. Although there are many types with slight differences in between, the major uses of simulations may be listed as follows. First of all, training simulations are popularly used to increase flight crew abilities and measure workload assessment. Secondly most of the aviation systems including avionics are simulated at early phases of development projects to decrease the costs by previewing potential faults. Lastly, to plan efficient aerial operations, models of the operational scene are very often used including “blue and red” aircraft teams in the presence of surface-based forces. Notice in cases where enemy or friendly aircraft are flying under the control of a computer, there is a practical need for computerized intelligence to create pilot behaviour to control the aircraft. The second chapter will provide a summary of previous studies that focus on pilot modelling, and motivate the need for better understanding the information flow within a cockpit.

The third chapter will review related cognitive modelling approaches excluding control theory based models that are already given in chapter 2. The modelling primitives employed for constructing artificial pilot cognitive abilities may have multiple forms depending on the aim and the methodology selected. The third chapter describes the related cognitive modelling work in the present literature starting from the symbol processing based cognitive architectures such as ACT-R and SOAR as well as connectionist approaches to modelling pilot behaviour. On the final paragraph of first part in this chapter, control-theoretic cognitive autopilot studies will also be mentioned and contrasted to symbolic and connectionist approaches. Finally, neural modelling primitives important for understanding DFT models such as conductance based single neuron models, including spiking and integrate and fire neuron models and finally the network based bio-inspired cortex modelling using single neuron models, will also be summarized in this chapter.

As previously mentioned this thesis aims to form a DFT based pilot model, so as expected it contains multiple aspects of dynamical theory and uses various older studies from the pioneers of this area. Dynamical field theory uses the dynamical neural fields (DNF) to model cognitive processes, especially spatial task behaviour including perception, task sequencing, memorizing and motor control. The second part of the third chapter provides information about the Amari equation as the roots of DFT, the philosophy behind dynamical modelling, the mathematics of used equations, information on previous works and proven applications, robotics applications and present frameworks available for constructing DFT architectures or models. In the latter part of the third

chapter a comparative analysis of the differences between DFT and other approaches are summarized.

The fourth chapter presents the architecture of the developed model, the structure of each sub-model and the integration of sub-models to construct a unified pilot. Additionally a 3 degree of freedom (3 DoF) aircraft model to be used in conjunction with the developed pilot model and the environment the simulated aircraft flies are also described in that chapter. Notice due to the dynamical structure that the DFT approach makes possible, the environment is also aimed to be dynamical so that the fast adaptation behaviour of DFT can be made observable clearly and functionally. On that thesis, the environmental effects such as atmospheric conditions will be generated and the outputs of the environment model will be used in conjunction with the 3 DoF aircraft model to create a dynamical behaviour. An iterative approach to model building was employed in this study, and the three models that correspond to three iterative stages are described in this chapter. The third and the most successful version is a pilot model with the capability of altitude awareness that can increase the vertical position of the aircraft during a hold performance task.

Finally, chapter 5 provides an evaluation of the model by comparing the flight records of a human agent and the performance of the most improved iteration of the DFT pilot model. The discussion will also include comparisons between various iterations of model pilots. Some of the difficulties observed with building DFT models in the aviation context and findings on the capabilities and promising aspects of the DFT approach will also be listed in this section. The thesis will end with pointers for future work.

CHAPTER 2

2 MODELLING AND UNDERSTANDING STUDIES OF PILOT COGNITION OR PILOTING TASKS

There are mathematical models of pilot abilities to control the aircraft from the aerospace engineering and mathematics area starting from 1950s. But those are functional models that just provide a transfer function constructed upon the control theory findings such as *root loci* and *Bodé* plots mostly based on the data acquired from questionnaires filled by pilots or from the flight test instrumentation. The term transfer function defines a mathematical relation between the input and the output of a system or a subsystem in control engineering. These models using such functions are focused on performing a specific function via using a pilot transfer function to control any part of the aircraft attitude. All such models use the pilot as a set of control equations in the control loop of an aircraft function. In this section brief summary of pilot models that can perform a spatial task will be given. Such models aim to create a pilot-like autopilot function which can simply be named as “control models”. Although those are not cognitive models of human behaviour, control theoretic models are still computational models aiming to simulate human behaviour, so we think that they are worth mentioning here.

In addition to the control theory based models mentioned above, in the literature there are also some rare studies that aim to model whole or some parts of the pilot cognition with various methods. We divide such methods in two parts, one being the “information dynamics models” and one being the “symbolic architecture models”. The second class of models will be discussed later in section 3.1. While making a classification between cognitive modelling theories using Fodor and Pylyshyn (1988) classification as a reference; we shall point out that most of the studies covered in this thesis can be grouped under representationalist camp.

2.1 Control Models

Rather than being tools for understanding human behaviour, control models are mostly intended for controlling the aircraft as part of an autopilot system and are the outcome of great engineering efforts. The very first such pilot models are mostly focused on determining the control parameters of vertical take-off and landing (VTOL) aircraft in the United States Air Force Flight Dynamics Laboratory (AFFDL) (Blakelock, 1991). The

purpose of this study was to determine the VTOL specification for a future development aircraft with a focus on emulating the human performance obtained in conventional aircraft. Since there were no such VTOL aircraft yet AFFDL did not have any chance to use real flight test data, so researchers preferred to use a simulated environment and obtained the pilot parameters via responses on the Cooper-Harper rating scale. The results led to the first control theory pilot model in the second part of 1960s, which have continued to serve as a resource for later studies such as the Johnson and Pritchett's (2002) generic pilot model. In the following paragraphs, although they are not the primary focus of this thesis, some introductory information will be given on the control theory based models based on Blakelock (1991), in an effort to motivate the DFT-based pilot models.

2.1.1 Crossover model

Crossover model is one of the two well-known and commonly used control theory based pilot models. The crossover model takes its name from the crossover frequency which is the frequency where the phase angle of a Bodé plot is -180° i.e. $-\pi$ (Blakelock, 1991). Bodé plots are very common tools of control engineering where the frequency domain characteristics of a system is plotted with axis organized as power of 10. The crossover model assumes a simple control loop of pilot-aircraft similar to one given in figure below.

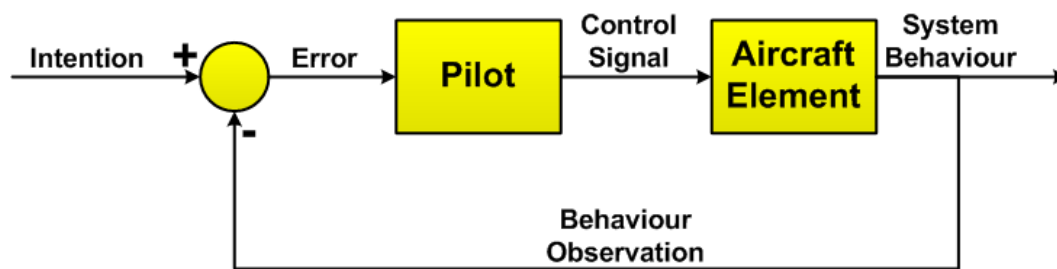


Figure 1. A representation of Crossover Model pilot-controlled aircraft element loop given with a slight modification from Blakelock (1991). Original drawing prefers giving control parameters in “s” domain. Here for ease of understanding signals and blocks are named with daily life terms in spite of using transfer function parameters.

McRuer and Jex (1967) give a crossover model written in frequency domain with parameters $j\omega$ and Blakelock (1991) provides the same model in Laplace domain with variable s . The terms $j\omega$ used in original study shows the transfer functions and all other blocks contributing to control are written by their frequency characteristics where the term ω symbolises a multiple of crossover frequency ($2\pi f$). But Blakelock writes the same function in Laplace domain where the system may be linearized using Laplace transforms and s is used as a symbolic input for any function. Neglecting the differences created from the model domain, in both cases, the blocks named as pilot and aircraft element (the parameter which will be controlled) are all modelled with transfer functions (a transfer function is the mathematical relation between the input and output of a black box block). Namely, the pilot transfer function is $G_p(s)$ and controlled element's one is $G_c(s)$. On crossover model's pilot-aircraft loop, we named the input as intention in above figure 1. The term intention may seem odd for a mathematical model, but to be able to create a relation between the cognitive meanings of each block in the model, using human behaviour context may be useful. At the end, this input represents the attitude value that

the model is aimed to hold by the implementer, so it is acceptable to name it as the intention or the goal. It is more preferable to use the term intention or at least the instantaneous intention since the term goal may cause a misunderstanding due to common usage of the word for ultimate or final purposes. In fact such intention may be a navigation goal originating from the flight plan or just a leg of a manoeuvre, whatever the case is, the pilot is mostly interested with the deviation between the intention (or instant/present goal) and the current situation. This is the main reason behind the stability of the crossover model. The model is working using the deviation between current situation and the goal. Below equation set gives a very simplified summary of crossover pilot model.

$$G_p(s) = \frac{S_p(\tau_L s + 1)e^{-s\tau_e}}{(s\tau_l + 1)} \quad (\text{Equation 1})$$

$$G_{mdl}(s) = G_p(s)G_c(s) = \frac{S_p S_c e^{-s\tau_e}}{s} \quad (\text{Equation 2})$$

On below paragraphs the parameters of the equations 1 and 2 are clarified. The model above is given in s domain as adopted from Blakelock (1991).

S_p : This parameter defines the pilot gain. In engineering applications the term “gain” is used to describe the multiplication coefficient for a parameter affecting the amplitude of its input.

τ_L : Pilot lead time constant. This is used to adjust the timely behaviour of the system. The next 2 parameters are also time constants. Such time constants are used to adjust the system behaviour’s similarity to human behaviour on timing aspects. Although the models seems a dynamical model including a negative feedback line for stability purposes, at the end it is clearly lacking the embodiment. Further discussion on embodiment issues on cognitive modelling are mentioned in various paragraphs on this thesis like 3.1.3 and 3.3.1.

τ_l : Pilot lag time constant.

τ_e : Pilot effective time delay. Blakelock (1991) defines this delay as the sum of the transport delays (τ_i) and neuromuscular first order lag (τ_N). Transport delays are defined as the delays in the central nervous system to achieve proper processing of the task.

Notice above model tries to achieve humanly dynamical behaviour with adjusting the time constants and neglects the spatiotemporal behaviour of the neural network. At that point, we can clearly note that control models of pilot are biologically irrelevant and just functional engineering models (probably very useful and challenging ones).

McRuer and Jex (1967) claim such model holds only around the crossover frequency of its Bodé plot. This is where the name of the model arises. Notice above given one is a very simple introduction to crossover model. Advance models such as second order crossover which can be used to understand/model pilot control over the phenomenal aircraft behaviour such as phugoids or Dutch rolls are neglected due to not deviate from our goal. Crossover models can be used on controlling the aircraft systems as autopilots using a total number of 3 transfer functions (Blakelock, 1991).

2.1.2 Paper Pilot Model

The Paper Pilot model belongs almost to the same era with the crossover model and in fact uses the McRuer and Jex (1967) model as a tool. It is a more complex model in comparison with the crossover model (Anderson, 1970; Anderson, Connors and Dillow, 1970). Figure 2 gives a schematic representation of the paper pilot model. Notice there are two loops of control, one being inside and in contact with the aircraft dynamics, and one being outside, namely the inner and outer loops. These two loop definitions are very commonly used in aircraft control modelling.

The paper pilot model use a performance based approach in the determination of the model parameters (Anderson, 1970). Usage of the transfer function from previous crossover model eased the development of the paper pilot. It is named as paper pilot probably since it is depending on the evaluation of the pilot performance using some questionnaires. In the model, most important parameters are the gain and the lead time constants, which are selected to maximize the model performance (i.e. minimization of the attitude errors). The following equation set defines the performance approach to paper pilot.

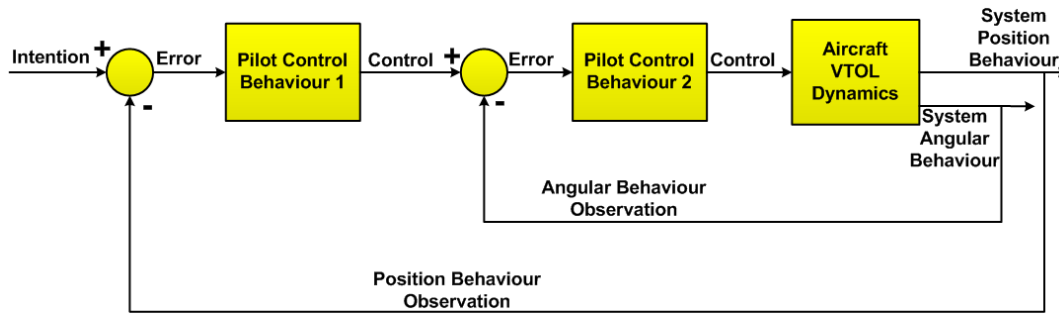


Figure 2. A schematic representation of the paper pilot. Notice the paper pilot different than the crossover pilot, understands not only the position but also the angular situation of the aircraft. To reach a positional goal, the model uses angular control. Above figure is modified from Blakelock (1991) where daily life names are not used but transfer function names are given.

$$PR = PERF + f(\tau_L) + 1 \quad (\text{Equation 3})$$

$$PERF = \frac{\sigma + \sigma_m}{\sigma_m}, \quad \text{limit @ 2.5} \quad (\text{Equation 4})$$

$$\sigma = \sigma_{\epsilon_x} + 10\sigma_{\epsilon_q} \quad (\text{Equation 5})$$

$$f(\tau_L) = 2.5\tau_{L\theta} + \tau_{Lx} \quad (\text{Equation 6})$$

$$\tau_{L\theta} \text{ Limit @ 1.3 sec}, \quad \tau_{Lx} \text{ Limit @ 1.2 sec}$$

On above set of equations, PR is the pilot rating function given by equation 3 which is used to evaluate close loop performance of the model. The term close loop is used to symbolise the system including its feedback lines. If the feedback lines would be

eliminated and only the feed forward system model is preferred, this would be an open loop performance. The term *PERF* given in the equation 4 is the main performance measure limited with 2.5. All σ terms with various indexes are standard deviations where σ_m is the desired model standard deviation for achieving required performance in equation 4. In equations 3 and 4, σ is a generic standard deviation applicable to all model as a function of σ_{ex} and $\sigma_{e\theta}$, one being standard deviation for the error in position control and the second being standard deviation of the angular rate in rad/sec. Finally $f(T_L)$ given in equation 6 is the pilot work load function computed using the position control lead time constant T_{LX} and angular control lead time constant $T_{L\theta}$ (Blakelock, 1991).

The Crossover Model described previously was able to control a single parameter, and controlling multiple parameters instantaneously requires the use of multiple crossover models. The paper pilot model has an advantage in comparison with the crossover model, since it uses both position and angular control to control the displacement. Note that this is the real situation in an aircraft where the pilot generally controls the angular movements and the thrust level (i.e. speed for displacement), so the paper pilot is far more realistic than the cross over model. Following the first version of the paper pilot, some modifications have been proposed on performance rating functions without changing the overall logic underlying the model. Additionally there are some stability augmentation system trails in the literature depending on the Crossover Model (Blakelock, 1991).

The main weakness of the paper pilot is its original performance dependency. Since it is performance dependent (i.e. the model parameters are adjusted by using the model performance iteratively), the model is very much dependent on the aircraft model, i.e. the pilot model performance changes with the aircraft model. So the model captures a task dependent pilot different than a human pilot who can learn to fly any type of aircraft with a performance depending on training hours.

2.1.3 Optimal Pilot Model

Optimal pilot model is built up to overcome the task dependency (dependency to aircraft dynamics) of the paper pilot. It is originally developed by Pollard (1975) in AFFDL. By its very novel nature, the optimal pilot model can be assumed as a very first implementation of the pilot's cognitive abilities, even though it is still a control model not necessarily based on biological principles. As it can be seen, from the crossover model to the paper pilot, the main improvement is made on the control loop ingredients with the addition of angular control, which is one of the primary controls performed by a human pilot. Similarly, the optimal pilot model goes one step further by adding a simulation of humanly behaviour in the form of an estimation-control loop. Figure 3 diagrammatically summarizes the structure of the optimal pilot model.

Similar to the Paper Pilot, the Optimal Pilot is also a model depending on a pilot rating function (PR) which is used to evaluate the model performance with current parameters like previously cited gain or time constants. One major difference between older model and the optimal pilot is that the neuromuscular or observation noises are aimed to be included in the model via noise input addition. White noise is preferred to make sure of that errors are distributed equally around the target value (Blakelock, 1991).

Optimal pilot model is tested against various aviation simulators and has shown significant success against various realistic 6 Degree of Freedom aircraft models. The dynamics of the aircraft as shown in above figure 3 are connected to the pilot model via a display system. Tests with original aircraft simulators of F-5, A-7 and Boeing-707 are also performed and reported successful on reaching intended performance (Blakelock, 1991).

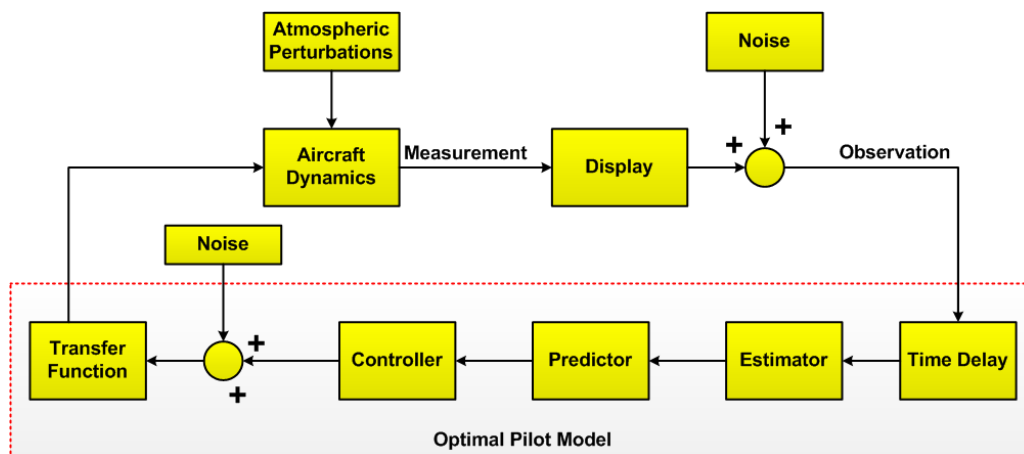


Figure 3. A schematic description of optimal pilot model is given. The blocks forming the namely “optimal” model are a series of transfer functions.

2.1.4 Generic Pilot Model

Johnson and Pritchett (2002) published a generic pilot model which can be used both as a simulated pilot in engineering and scientific applications and as an autopilot controller. This model can be considered as one of the newest and most generalized one among all control models. Although the Generic Pilot model is newer than the previously described model by three decades, it is also based on the studies of McRuer and Jex (1967). Before describing the specific properties of the generic pilot model, definitions for some of the fundamental concepts in autopilot systems will be given.

A pilot controls the aircraft with no direct contact with the position or attitude. The change in the position of an aircraft is the result of the speed, moments, inertia, current angles of the body etc. Similarly angular control of the body is not directly performed; instead the pilot controls the body moments via changing the positions of flight control surfaces. The control parameters in an aircraft control system can be divided into two layers; a layer of parameters in direct contact with the aircraft body, and a layer formed of higher level parameters much more on global level. In other words, this organisation is similar to Marr’s three levels of organisation where the body of the aircraft resembles the biological layer, the direct parameters resemble the representation layer which is in direct contact with the biological layer, and finally the indirect parameters are similar to the algorithmic level which form the pilot’s intentions or mission parameters e.g. climb to a specific altitude, fly to a specific location etc.

The above given description of high and low level parameters are grouped on two different closed loops in autopilot studies, namely the inner and outer loops. The outer loop includes the high level parameters which are transformations of pilot’s intentions

into aircraft control parameters. The inner loop is direct angular effects in the body of the aircraft. Below figure summarizes the relations between the inner and outer parameters.

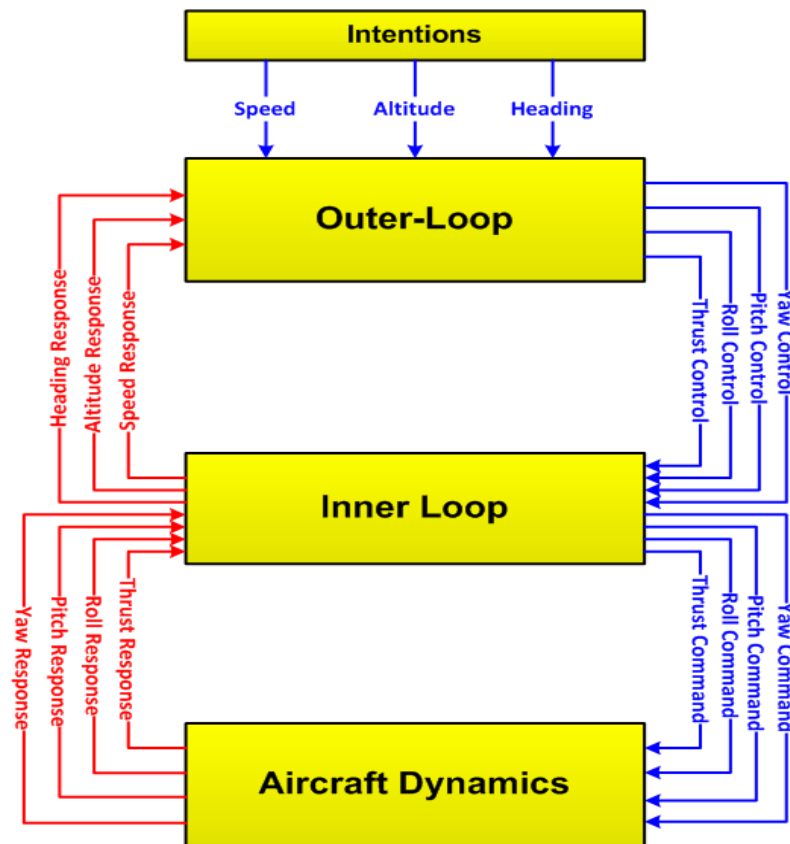


Figure 4. Inner and Outer loops of control are shown. The figure is a modification from Johnson and Pritchett (2002). Information flow towards the aircraft body is shown with blue and towards the pilot is shown with red lines respectively. Arrows point out the direction that the control flows.

2.2 Studies on Cognitive Information Flow in the Cockpit

The control theory based models given in the previous section as well as the DFT model that will be presented in subsequent sections provide mathematical characterizations of pilot behaviour. In contrast, the information flow models do not provide such a mathematical formalism. Although such models lack mathematical specificity, they still provide useful and valuable conceptual structures for understanding piloting from an information processing perspective. Despite previous and later models given in this study are generally mathematically based cognitive models and information flow models do not aim to provide any mathematical approach, the mentioned group of information flow models are still very valuable and information rich architectures. Even though the information flow studies mostly focus on civilian airliner type aircrafts, when one considers navigational and control capabilities, the mentioned models are also applicable to military aircraft since primary tasks are very similar. Hutchins (1995) defined a cockpit environment not isolated from the pilot but being an embodied environment for pilot cognition with a case study that documents how pilots use memory cards to remember

important speed limits. The acceptance of artifacts and the body as extensions of the human mind in specific situations is a very popular debate since the beginning of the last decade pioneered by Clark (with a very much discussed prologue by Chalmers) (2008). Hutchins (2010), even sometimes critical of Clark's extended mind thesis by claiming that it lacks social interaction (i.e. human to human extensions/interactions of minds), provides a similar approach to cognition and information flow in the cockpit a couple of years before Clark. Regarding this approach it may be proposed that Hutchins' models are not only pilot cognition models but larger scope information flow models in cockpit including the pilot. Hutchins (1995) in his pioneering study designs a relationship between the internal and external representations while summarising pilot cognitive workload during some specific tasks in his pilots' usage of aircraft weight-speed memory card example.

It is, rather, a combination of recognition, recall, pattern matching, cross modality consistency checking, construction, and reconstruction that is conducted in interaction with a rich set of representational structures, many of which permit, but do not demand, the reconstruction of some internal representation that we would normally call the "memory" for the speed (Hutchins, 1995, p.284).

The problem of defining the obscure relation between the external and internal realms of human cognition is a primary problem of cognitive modelling (Eliasmith, 2013). In another study, Hutchins (2000) classifies the flight deck as an interface between the flight crew and the artificial intelligent systems behind the deck. Hutchins argues that the airliner cockpit desk as a cognitive system including the pilot and the physical artifacts and examines the information flow inside the airliner; Hutchins parallel to his critics on Extended Mind Thesis, tries to show the information flow in the cockpit in between the humans and between the human and the machine forms a cognitive system where paths can be shortened and optimized for better pilot performance. A similar analysis is given in Hutchins & Klausen (1996) depending on the data derived from pilots' speech records during flight in a simulated environment.

This mentioned study shows there are various possible channels of information flow between the artificial intelligence (AI) and the crew and between individual members of crew, and showing the professional maturity or expertise of human members of this team may increase the quality and effectiveness of the flow and may eliminate possible failures. One important point to be underlined in this study is that Hutchins summarises the procedure of vertical descent rate computation made by a pilot under certain conditions describing the deck displays as just sources of information, and naming the mentioned computation as **disembodied computation** since it is done with a complete consciousness. Although Hutchins uses the term "disembodied" for conscious computations, it shall be notified that regarding the embodiment theory as used by dynamical field theory approach, all computations internal to human mind are embodied without considering the consciousness of the agent. What Hutchins tries to point out is clearly a difference between conscious mathematical computation and unconscious control action, both being cognitive tasks of the pilot, but it shall be noted the usage of the term "disembodiment" in the description of a computation's consciousness level creates a ambiguity. Despite its contradiction with the embodiment theory, Hutchins's claim of disembodiment of some processes when they are offline with the external world and

performed completely internal to the mind may be found plausible by a computationalist approach.

Although information flow models do not provide any identification on the biological mapping between models and the human cortex, they identify how a human cognitive system operates under heavy mental workload conditions in a sociocultural environment filled with complex artifacts. There are various studies that aim to distinguish how the communication and interaction inside a cockpit are mediated by some cockpit objects such as displays, dials, papers, autopilots or communication radios etc. Some examples may be research on human errors perspective in aviation by Wiegmann and Shappell (2000), information processing model for human-computer interaction by Ye (2003), distributed airliner cockpit cognition by Hutchins and Klause (1996), cultural effects in flight deck cognition by Hutchins, Holder and Perez (2002) and by Nomura and Hutchins (2006), the use of paper and its effect in flight deck cognition by Nomura, Hutchins and Holder (2006) and risk management in pilot cognition by Hutchins (2001). Hutchins (2000) also analysed pilots' learning behaviour using a term frequency and term occurrence method and gave quantitative results showing the training duration of airliner pilots is not enough to develop complete expertise on the use of complex autopilot modes in lateral and vertical navigation and most of this capability is developed only after starting the active mission flights.

Nomura and Hutchins (2006) made a survey comparing Japanese and US pilots in terms of the differences between their cognitive and interactional processes. The paper shows a linguistic improvement specific to the so called cockpit language, even including body language such as gestures. Parallel to this study, Nomura, Hutchins and Holder (2006) made a cognitive analysis of the use of paper material in a cockpit as a cognitive tool by pilots from multiple cultures. Hutchins (2001) made a survey of risk handling of pilot cognition in the cockpit using post flight interviews and speech records in flight. This paper shows evidence on the pilot being affected of social culture of her/his origin, e.g. American culture for American pilots.

Hutchins and Holder (2000) show that the pilot uses predefined concepts for understanding the "pop up" events occurring during a flight. For instance, the pilot is assumed to construct a concept in his mind for possible events (such as encounter with a mountain wave) and uses these concepts to analyse his current situation in the flight. One by one comparison of all inputs with each item of the concept may lead the pilot to make a decision. A similar study by Holder and Hutchins (2001) also claims that the pilots engage in "concept development" to learn the complex tasks in a cockpit (such as autopilot).

Notice whatever the node –a paper, a display or a human– that the pilot is interacting with, most of the studies mentioned above assume an interface of information which is processed by the pilot to create an action solution. The diagram in Figure 5 represents a generic architecture of information flow using the common issues of above mentioned models and studies.

On below figure reflecting Hutchins approach to cognitive architecture interacting with external artifacts, processing block is demonstrated working in conjunction with memory blocks. This blocks of memory whatever their nature or modelling method is, are very common in all cognitive architectures since memory function is an ultimate function of

human cognition. In the DFT model that is provided in this thesis, memory models are also used for supporting executive functions.

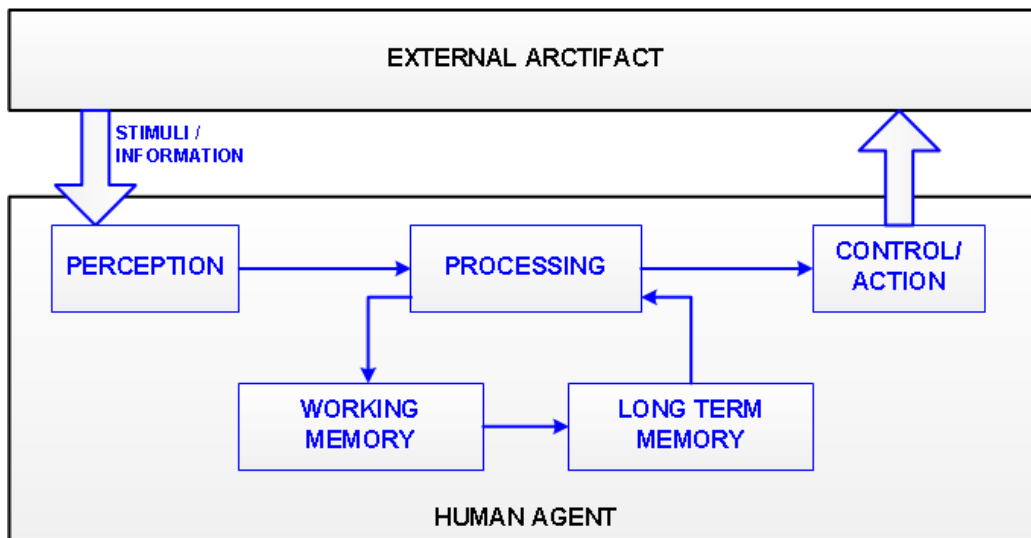


Figure 5. A generic model of information flow between human agents and external artifacts.

CHAPTER 3

3 COGNITIVE MODELLING APPROACHES

3.1 Symbolic Architectures

On the previous chapter some of the major control theory models for piloting tasks were summarized. As it can be seen, some of those models are task-specific and some of those, especially the relatively newer ones, are task independent models. Some of them are task dependent and some of them, which are in fact the famous ones like ACT-R or SOAR are generic ones. Our review will focus on special applications of generic cognitive architectures on piloting ability development. But firstly we will provide a summary of the definitions of the commonly used terms in the field. The following definitions are summarized from teachings of International Council of Systems Engineering (INCOSE), which is one of the largest institutions for system architecture development in the world.

An “architecture” is a schematic representation of the distribution of the functions on the components of a system and the relations (such as interfaces or connections) between those components. It can be developed on various levels, e.g. architecture of physical layer which shows the physical connections of system components or functional architecture which shows the functions and functional relations of the components. An interface defines the relationship, functional, physical or electrical, between more than one component of the system or between the user and the system or the components. Interfaces are properly identified on an architecture so that the system information flow is determined. The determinism of a system means that the system provides the same response under the cases where the input set is the same, i.e. if conditions do not change, outputs do not change.

Since its inception, the Connectionist approach to cognition argues that intelligence arises from the network (Fodor and Pylyshyn, 1988). Oppositely, cognitive architectures, even the connectionist ones, are formed of functional blocks cooperating to form cognition. Anderson and Taatgen (2009) trace the origin of the idea of cognitive architectures back to Alan Turing (1950) and the Turing machine, credit David Marr (1982) as the champion of the idea, and name the General Problem Solver (GPS) of Newell and Simon (1963) as the very first example of machine generated cognition. Marr’s idea of algorithmic abstraction between the biological medium and observed behaviour is considered as the idea behind the functional architectures of cognition. While Turing lists the memory and

processing limits (i.e. physical and time limits) as the only limits for the machine intelligence, current situation shows that such ability is a challenge to endure all efforts of scientists and engineers (Anderson & Taatgen, 2009). Simulating the functions of the modules of the human mind can be a step towards the understanding of the overall details of the function and behaviour of humans. So one promising method is to construct functional architectures to simulate human behaviours, even in cases where they are not biologically based. Cognitive architectures have emerged as strong tools to realize these ambitious goals.

General Problem Solver has been a source for the SOAR (Laird, Newell and Rosenbloom, 1987) architecture, which is one of the seminal architectures, whereas a study by Anderson and Bower on the human associative memory (HAM) has been a source for another architecture named ACT-R (Adaptive Control of Thought-Rational). The above mentioned ACT-R and SOAR are among the most widely used architectures in the cognitive modelling literature. Other architectures include EPIC, CLARION, 4CAPS (Wray and Jones, 2006). In the following paragraphs, we will summarize only the first too, since they are the only architectures used for the purpose of modelling piloting tasks.

3.1.1 ACT-R

Anderson, Bothell, Byrne, Douglass, Lebiere and Qin (2004) describe ACT-R as their best hypothesis about the unified theory of cognition architecture requirement that they quote after Newell's (1990) best hypothesis being SOAR. According to the mentioned study, coherent cognition will only arise with the completely integrated functioning of all modules or parts of the mind as a whole system. ACT-R is cited as an architecture promising a framework to model cognition partially or completely depending on how it is used. ACT-R is abbreviation for Adaptive Control of Thought-Rational and in this document the version 5.0 of the architecture is observed. In another book chapter Anderson recites Simon's ant analogy to describe the main idea behind ACT-R (Anderson, 2002). Simon's ant is tracing a very complex navigational route while walking in the sand beach but in fact an ordinary ant is a simple cognitive creature and the complexity of the route is caused by the complexity of the surface that the ant crawls. Anderson makes an analogy between the ant and the human cognition referencing Simon while claiming human cognitive processes are in fact basic and the resulting complex behaviour is due to the complexity of the human cognitive knowledge stored (Anderson, 2002). Parallel to this approach, ACT-R tries to provide framework to model complex human behaviour via a basic architecture.

ACT-R is formed of modules and buffers responsible from connection (acquisition of input and retrieval of output) between modules and the production system. The production system (shown as the production block) is where the systematic action selection is performed like the human Basal Ganglia and formed of three sub-parts, namely: Matching, Selection and Execution. Modules contain blocks responsible from intentions, declaration (more than a memory function), visual and finally the manual (or motor) modules. The complete architecture is connected to the external world via the visual and motor modules (Anderson et al., 2004; Anderson, 2002). On below diagram a schematic representation of ACT-R v5.0 is given.

ACT-R is probably the most well-known and commonly used one of cognitive architectures and one can find multiple applications or experiments of aviation tasks implemented in ACT-R. Even Taatgen and Anderson (2009) report and implementation

of ACT-R in a neural network despite existing general acceptance of connectionist approaches opposing symbolic architecture claims. One very attractive side of ACT-R is that various modules can function parallel as in a human agent (e.g. visual acquisition continues while some process takes part in basal ganglia) thanks to ACT-R providing independent buffers resulting in the possibility of mentioned parallelism. In the opposite way, each buffer may only contain a single unit of information (namely a chunk in ACT-R community) so that a module can provide only a single data in a single step of operation creating a strong serial queuing similar to human agent memory retrieval. One very positive property of ACT-R is that it provides a learning function both in statistical and structural manners i.e. ACT-R even may learn new rules adaptively ending with a modification of production (Langley, Laird and Rogers, 2008).

Such properties make ACT-R a smart tool providing good correlations in many cognitive tasks modelled (Anderson et al., 2004). An aviation related experiment and model set cited in above reference contain the model of a RADAR operator performing anti-air warfare system control providing acceptable detection timing and motor performance correlations between the agents (model and human). Langley, Laird and Rogers (2008) report ACT-R being used in commonly in some tutoring systems in schools and in human-robot interaction works.

Above reference gives some common capabilities generally contained (and of course in some applications desired) in cognitive architectures. These are reported as “recognition and categorization”, “decision making and choice”, “perception and situation assessment”, “prediction and monitoring”, “problem solving and planning”, “reasoning and belief maintenance”, “execution and action”, “interaction and communication” and finally “remembering, reflection and learning” (Langley, Laird and Rogers, 2008). ACT-R is covering above capabilities depending on the application. The reader may refer to Langley, Laird and Rogers for a detail evaluation of various architectures which are not mentioned here such as EPIC or CLARION (2008).

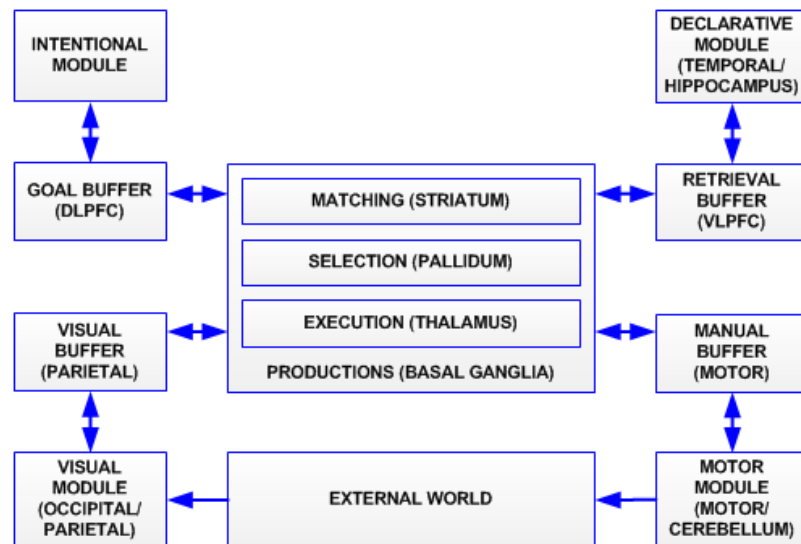


Figure 6. Overview of ACT-R v5.0 (Andersen et al., 2004; Anderson, 2002). A similar architecture is given for ACT-R v6.0 in Taatgen, Lebiere and Anderson (2006).

3.1.2 SOAR (States, Operators And Reasoning)

SOAR is intended to focus on the cognitive layer of human mind omitting the neural level and assumed rational level above the cognitive layer (Rosenbloom et al., 2002). Starting from the assumption of that the cognitive layer supplies enough capability to model human cognition without the neural level, with two other assumptions a basis to SOAR is formed. Secondly it is assumed that artificial agents can be accepted as intelligence as human agents. In fact such assumption shall be valid for all other artificial intelligence studies naturally. Third, it is accepted that all complex intelligence shall be constructed from smaller and relatively basic parts and development of such parts may aid in construction of complex behaviour. Lastly architectures may evolve towards the desired goal of general intelligence (Rosenbloom et al., 2002). Clearly this is also a default acceptance for all ongoing studies of artificial intelligence.

SOAR is a goal (or intention) oriented architecture formed of levels. The first of levels is the memory since all intelligence requires an amount of memory. SOAR prefers to hold all knowledge without categorizing them as declarative, episodic or procedural, in a single long term memory structure. A global memory level is hold above the long-term memory and all information retrieved from long term memory is transferred into a global working memory. Traditional logic approach or classical production system (here classical may refer to what is done in ACT-R) do not apply to SOAR. In SOAR or productions are parallel retrieval operations from long term memory. So there is no conflict resolution requirement in SOAR. Despite described different structure it is possible to focus on logic or classical production operations by implementing them in long term memory storage (Rosenbloom et al., 2002). In one stores a declarative knowledge in SOAR memory level, then when necessary all knowledge can be acquired to the working memory. But if the knowledge is type of procedural as logic operations or production operations, then it cannot be fully retrieved to the working memory but rather it is executed.

An architecturally provided fix decision process or procedure forms the decision level of SOAR. On the first part of the decision procedure, all memory accessible and relevant to current decision is accessed and actions are listed. Notice none of the relevant information is omitted during this retrieval. On the second part of the procedure correct action is selected and executed. The selection of the action inside the procedural memory storage can be either plan-based or reactive. Under normal conditions, each problem solving decision is based upon the action or logic productions stored in long term memory and the decision is selection of a problem space, a state or an operator. All the time that the memory stores correct knowledge SOAR performs correct decisions being algorithmic or logic. But in cases where the knowledge is not correct, the decision may lead to incorrect decisions. Even when there is correct knowledge, decisions due to algorithmically unpredicted cases (such as weak systems engineering in an expert system) the decision may lead to incorrect decision. For detectable cases it is possible to make a backward run or create an impasse.

Finally below all levels of SOAR, there is a goal level. Goal levels depend on the decision levels results. Goal level is activated to set a new goal each time there is no preference between equally valid productions (tie) or there is no production (no-change) left in decision level. These special situations are called as impasses in SOAR area. Each time such an impasse situation arises, than a new goal and performance criterion assessed with the new goal is set. So setup of new goals can be a continuous process until the

moment that there rise no more impasses preventing SOAR from shaping any more goals (Rosenbloom et al., 2002). In fact this is the most promising part of SOAR, it is goal oriented and solutions originally are bonded to goals. Probably this is the aspect closest aspect of SOAR to a human agent.

For an implementation of a task in SOAR, required knowledge is formed of the initial state of the variable, the final or desired state and the operators. Notice this set also forms a problem definition: “*reach stated f starting from state i using the operators +, - and /*”. This setup is originally named as the Problem Space Computational Model (PSCM) and is the heart of SOAR used in performing computations (Wray and Jones, 2006).

Regarding the above descriptions, it may be accepted that one of best properties of SOAR is that it is perfectly goal oriented while running a knowledge based task or tasks list. If not limited by the hardware, in any area SOAR capabilities is limited with the knowledge limit of the developer. For task specific intelligence studies, SOAR may provide the user with full capability of parallel retrieval from procedural and propositional (logic) productions storage.

Although there are various applications using SOAR for development of cognitive models with specific functions such as interactive computer game character intelligence, illness diagnostics or human language processing such as Cypress-Soar and Designer-Soar algorithm design systems, Neomycin-Soar medical diagnosis system and Merl-Soar job-shop scheduling system, probably being the most famous application and aviation related, TAC-Air-SOAR shall be mentioned at that point (Langley et al., 2008; Rosenbloom et al., 2002; Wray and Jones, 2006). TAC-Air-Soar is a tactical fighter simulator running on soft real time based on more than 8000 rules developed by aerospace experts held in the long term memory in form of procedural knowledge. Notice different than the scope of this study, TAC-Air-SOAR is not focused only the embodied navigation piloting function but rather includes many decisive functions normally under the responsibility of a human pilot. TAC-Air-SOAR provides a very capable pilot model performing formation flight even in lead position, communicating with other aircraft, and even can change the content of its intended mission during a flight and orients towards another one. Probably it is the most developed expert system model developed in cognitive architectures due to its enormous content of expert area knowledge.

3.1.3 Embodiment Critics on Symbolic Architectures

Although there are various other cognitive architectures in the literature available for scientific research, ACT-R and SOAR are the most well-known and massively used ones. The main segregation between the definitions of these two architectures is that ACT-R is much more addressed towards biological tissues and functions in comparison with SOAR, resulting in a spurt of model performance comparison experiments with human agents (Eliasmith, 2013). Even there are critics on the biological plausibility of ACT-R modules; our primary concern is not to evaluate the symbolic architectures. Instead of focusing on the continuing debate between the symbolic or representational architectures and the connectionist approaches, in this section we will provide a very brief review of the literature that evaluate such architectures from an embodied cognition perspective.

Clearly symbolic architectures are much more focused on the symbolic manipulation considering the semantics embedded inside the syntactic operations. Naturally this property is not different than the connectionist approach but the main limitation of the cognitive architectures are the neglecting of the neural or biological level. Although there

are cognitive models based on architectures like ACT-R that successfully account for response time characteristics of human subjects, these models are far from being neurally based directly. In section 3.3.3 some temporal properties of human cognition in form of dynamical system will be provided. In that section, it is shown that, with ACT-R it is possible to implement cognitive functions such as an internal clock which is biologically false with great probability (Spencer, Karmarkar and Ivry, 2009; Eliasmith, 2013). Eliasmith (2013) has also made a strong criticism of the ACT-R timing behaviour as the architecture entirely neglects neuronal firing rates. The assumption of having an abstract cognitive layer functioning independent of the biological tissue is far from being realistic, even though it may be sometimes useful in understanding certain mechanisms of cognition.

Probably, in the long term, parallel to Anderson's pointed requirement of a unified theory of cognition, symbolic and sub-symbolic architectures may evolve to be unified under some hybrid architectures. By defining an architecture or framework with capability to model the abstraction layer (of ACT-R) or the cognitive layer (of SOAR) with embodied infrastructure containing neural firing rates or timing behaviour, an promising tool for cognitive modelling may be at hand. At least it may provide beneficial and plausible aspects of previous approaches under one flag. With such an architecture, we may pursue the task independent intelligence issues with cognitive models in a more satisfying level. In the final paragraph of this thesis, the difficulties observed while using DFT for complex functions modelling is given. A special tool unifying advantages of symbolic architectures with the embodiment advantage of sub-symbolic modelling will be very beneficial on overcoming the difficulties encountered during mentioned complex functions modelling.

3.1.4 Neural Network Based Pilot Models

In previous subsections we reviewed some control theory based models and some symbolic architectures. Another group of models contain connectionist approaches to implement neurally based pilot models. Those are obviously much closer on being cognitive since they are depending on neural networks. Some examples can be read on Enns and Si's (2004) helicopter flight control with Neural Dynamic Programming, Kaneshige and Burken's (2008) neural network based in-flight control model of a F-15 aircraft in real flight tests, and neurally informed intelligent models developed in NASA (Motter, 2008). Due to the charming success of these models on control of aircraft, the critics provided here will not be on the aspect of performance but we will underline some conceptual differences between connectionist and the DFT-based models here. Mainly those models do not provide any embodiment due to usage of classical mathematically based neural network approach and none of them uses neural fields to implement the network. On this aspect the DFT framework is completely different from previous studies.

Although the approach and models are completely different from previous neural network based models, on the implementation of our DFT based pilot model, some properties of previous models are also used. In the below paragraph, such inherited properties are summarized.

The DFT based pilot models given in this study use an intention layer to supply the pilot model with a goal. This layer is shaped by rising a self-excitatory dynamic neural field with a very low decaying time constant so that after rising of the field activity, due to very

high time constant decaying rate, the field does not fall to resting level even in lack of the input resulting in that the pilot model does not forget the memorized spatial value. For further details of memory functions please see paragraphs 3.3.2 and 3.3.3. Using a set of reference fields or models to create an intention is first given in the study by Kaneshige, Bull and Totah (2000), where 3 separated channels are used as reference models for pitch, roll and yaw rates of the aircraft. Dividing the intention or the memory holding the intentions into 3 components as reference models of behaviour provides a useful and easy to implement method. We have similarly used 4 segregated neural fields that hold or “store” the pilot’s various intention patterns, which will be described further in subsequent sections.

An important difference shall be noted here; Kaneshige, Bull and Totah (2000) model uses neural networks with s-domain transfer functions to control the aircraft, so no neural fields are used in this approach. The models given in our study will include time domain functions while the cited reference performs the control modelling in s-domain. While Kaneshige et al. continuously compare the reference models with current rates and form the next action based on their difference, this loop can be assumed as a negative feedback line to achieve stability similar to the control theory based models given in section 2.1.

To sum up, existing pilot models in the aviation industry are predominantly control theoretic models that focus more on control dynamics to achieve control performance rather than cognitive processes underlying a human pilot’s performance. In this study, we describe a Dynamic Field Theory based approach to extend this line of work by incorporating embodied layers that modulate cognitive processes underlying pilot behaviour without giving any biological mapping. The next chapter provides an overview of the principles and mathematics of Dynamical Field Theory. This is followed by a description of the simple aircraft model used in this study and the 3 levels of DFT-based architectures developed to model pilot behaviour. The final chapter of the study provides a validation of the DFT-based pilot model in the context of an altitude hold task in reference to real human pilot data.

3.2 Neural Models and Bio-inspired Models

Before starting the discussion of neural modelling, a set of definitions for the terms neuron, cortex and relevant issues might be beneficial since some of the modelling terminology includes direct references to the biological tissues or components. The term component might seem unusual for biological units, but in fact regarding the neural system as a system of systems, it is acceptable to abstract the whole body under different blocks of sub-systems.

There is no single type of neuron in human neural system. In fact there various types of neurons (pyramids, baskets, Purkinjes etc.) with specific capabilities (sensors, motor processing, emotional etc.) and structural properties, but some properties are solidly common in all types of neuron (Venkov, 2008). A neuron for an engineer and modeller is a system or component with a very complex and dynamical transfer function, and so while having a transfer function; in fact it has an input(s) and a relevant output. These input ports in a neuron are called dendrite and the output port is called axon. Despite existence of multiple input ports, in fact there is only a single output port. Between these inputs and the outputs there is the cell body (namely soma) containing the cell nucleus. Although there are other components in a neuron such as Myelin Sheaths (covering the

axon main body) or Ranvier Nodes, for mathematical modelling purposes those are simply negligible. On the below drawing a simple schematic for a single neuron is given.

Across the dendrites and the axon, there are electrochemical behavioural differences parallel to the functional differences. Dendrites work continuously and sum up all the input gathered from neighbouring neurons. Generally the dendritic transmission function is assumed linear and proportional to the length of the dendrite. i.e. if an input belongs to a neighbour neuron which is relatively far from the subject neurons soma then the input from that dendrite will decay with distance (this is similar to the distal inhibition effect which will be discussed in later paragraphs). Completely opposite to the linear and continuous behaviour of the dendritic currents, the axon behaviour is impulsive (Venkov, 2008). This means that the axon does not produce a level behaviour continuously, but provides impulses with a frequency of activation depending on many parameters. Each axon output is collected by the next neurons' dendrites with interface layers which are called as synapses.

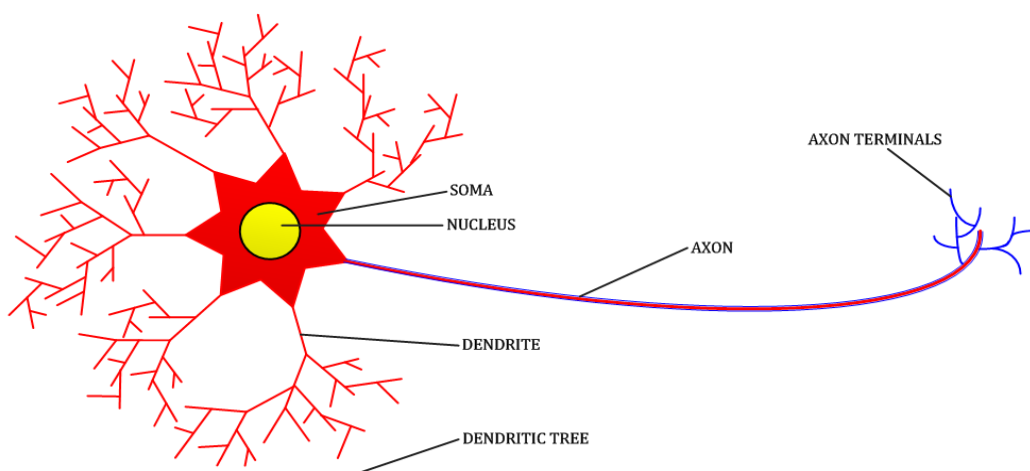


Figure 7. The schematic representation of the main parts of a neuron.

The axon's presynaptic behaviour of impulse production provides a transmission medium with an electrical activity media to be collected by the postsynaptic dendrites. Notice even the axon produces impulses of electrochemical activity, the chemicals of the neural system diffuse from the borders of the units with ionic concentrations and so the chemical behaviour provides continuity in the cortex. Notice the diffusion is not a fast mechanism in comparison with the technological transmission methods. Bear, Connors and Paradiso (2001) provides that the electrical data in human neural system travels with 10m/s of transmission speed.

In the diagram given in figure 8, the dendritic currents are assumed to be summed up with weights proportional to the length of the dendrite, but notice there can be odd cases where the dendritic tree adds an increasing effect to the amplitude of the neural current due to the behaviour of the synapses, which is called excitatory behaviour. Different from the naturally decaying dendritic current, an axon response or transfer function can be assumed productive, i.e. can increase the amplitude in case of some activity threshold is passed.

Neuron electrical behaviour is a very complex function and it is not easy to model with only one direction modelling. One of the very first reasons of the neuron complexity is

the oscillating behaviour of the output due to the feedback characteristics of the cell membrane (i.e. reverse electrochemical diffusion). i.e. a neuron's own products at the end limits the output of the neuron and causes an oscillating output. Additionally the dendritic tree is very complex and each neuron has different dendritic characteristics. Although under such circumstances an accurate and complete neuron model is not achieved yet, still there are functionally very useful models.

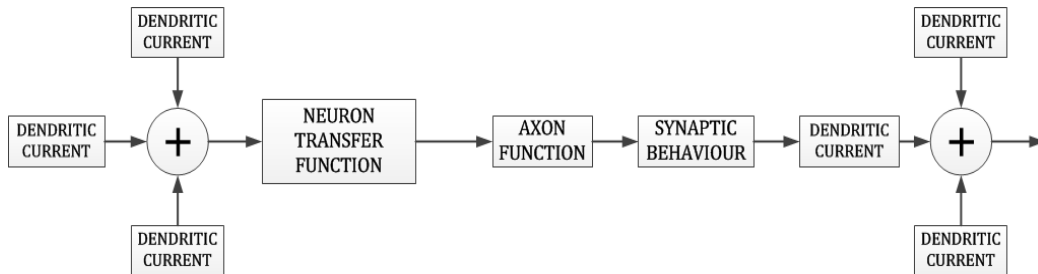


Figure 8. The representation of the neural electrical transmission direction and functions from a neuron to the next one.

After achieving such a neuron model, one can implement networks of such neurons to produce cortices or nuclei. The network may be bio inspired or not, depending on the function and the purpose of the study.

A cortex is most of the time a “plane” on the surface of the brain, being planar meaning isotropy in the planar directions. In other words, a cortex (with exceptions) is a plane where the one direction along the plane holds geometrically same neural organisation with the other direction of the plane. Such organisation is achieved with using columns vertically organised along the depth of the cortex. The thinnest columns are called microcolumns and they are unified to form macrocolumns. Binding together the macrocolumns vertically aligned, one can achieve a cortex. Even there are cortices unsymmetrically shaped in 3D, most known examples of cortices (such as hippocampus – simply memory- and cerebral cortex –intelligence-) are mostly in laminar form and are isotropic in plane. Laminated structure of cortices gives rise to the idea of those layers might have differing functions (Venkov, 2008).

A nucleus is as expected different than a cortex is isotropic in 3D and does not have to be a surface or a laminated plane. A Nucleus is a 3D network of neurons that are isotropically organised and focused on a special function. A much known example of nucleus can be the Lateral Geniculate Nucleus (LGN), a part of the visual function of the human brain, which functions on the 3D monocular processing of the binocular 2D input from eyes.

3.2.1 Conductance Based Models

As described above, a neuron model is generally approximated as an electrical device where sum of input currents and output are related via a function. Most of the time it is assumed that a neuron's non-linear behaviour is similar to the charge-discharge oscillation of a capacitor. So a neuron can be modelled using a capacitor current equation in a very primitive way. The current in a capacitor is equal to the time derivative (rate of change) of the voltage between the input terminals and a capacitance coefficient C .

$$C \frac{dV}{dt} + \sum_k I_k(t) = I(t) \quad (\text{Equation 7})$$

In the above equation 7 the $I(t)$ is the synaptic currents. Assuming that the synaptic current is input to such a system, the current can leak through the ion channels ($\sum_k I_k(t)$) or some part will charge up the capacitor ($C \frac{dV}{dt}$). As it can be seen from the above equation 991, in fact it is a very simple approximation. Yet the true behaviour changes regarding how the leakage currents are modelled and how many channels are used. Such models are named as conductance models.

One very famous capacitance/conductance model is the Hodgkin-Huxley Neuron Model (HH Model) (1952). In that model, the ion channels are divided into 3 categories (Na^+ , K^+ and cell membrane leak) and in each category conductance is shaped non-linear proportional to the applied voltage for ions and independent from voltage for leakage. So the ion and leakage current of the HH model can be given with the below equation:

$$\sum_k I_k(t) = g_{\text{Na}} m^3 h (V - V_{\text{Na}}) + g_{\text{K}} n^4 (V - V_{\text{K}}) + g_{\text{L}} (V - V_{\text{L}}) \quad (\text{Equation 8})$$

In the above equation 8, the terms of ionic currents are in form of $[g_{\text{Na}} m^3 h (V - V_{\text{Na}})]$. Here g is the conductance constant for a specific ion type. The m , h and n are called gating variables and those are values between 0 and 1 describing the probability of the ion channel to be open or closed. V_{Na} , V_{K} or V_{L} terms are the reversal potentials forming a threshold for the oscillatory behaviour. Each of the gating variables is a non-linear function with specific time constants functions.

$$\tau_m(V) \frac{dm}{dt} = m_{\infty}(V) - m \quad (\text{Equation 9})$$

$$\tau_m(V) = \frac{1}{\alpha_m(V) + \beta_m(V)} \quad (\text{Equation 10})$$

$$m_{\infty}(V) = \frac{\alpha_m(V)}{\alpha_m(V) + \beta_m(V)} \quad (\text{Equation 11})$$

In Figures 9 and 10, the gating variables and time constant functions are sketched with input voltage at horizontal axis.

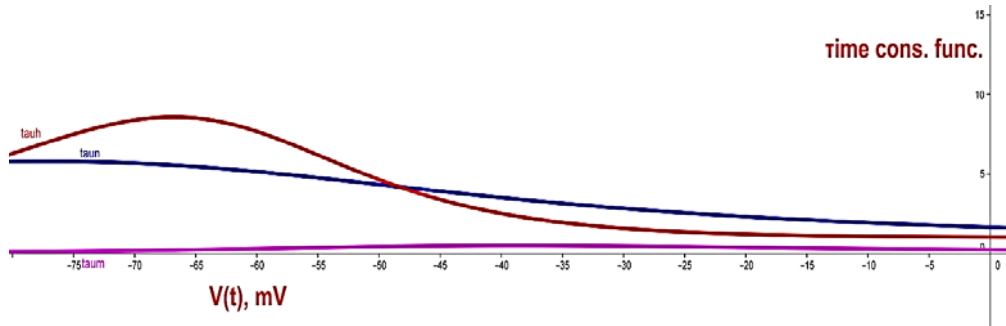


Figure 9. Time constant function amplitudes representation for gating variables of HH model. Notice the location difference of peak values for each time constant forming at a different location (reverse potential points). This graphic shows how the time constant of

each gating variable changes regarding the $V(t)$ value. Red line is the time constant for h , blue is for m and cyan is for n .

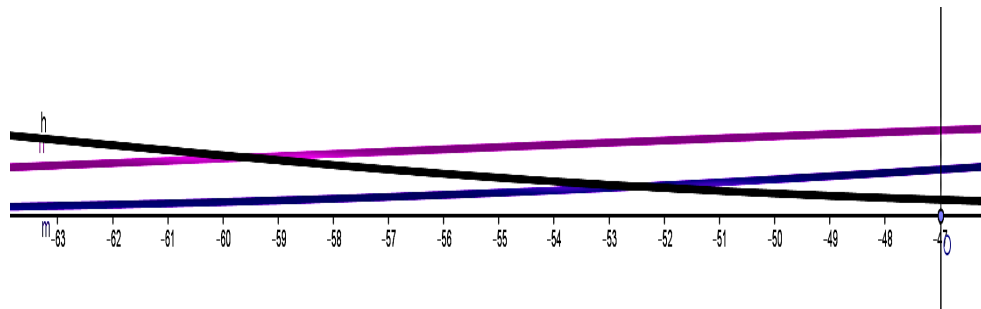


Figure 10. Gating variable representations for gating variables. This graphics shows the changes in the amplitude of gating variables with respect to $V(t)$ value. Red line is the time constant for h , blue is for m and cyan is for n . Notice h decreases while m and n increase.

As it can be seen in above equation set, the time constant and the maximum value of the asymptotic gating variable m are all non-linear depending on the applied voltage. Two similar set of equations (14, 15 and 16) shall be written for the parameters h and n . Notice the Na^+ channel voltage is multiplied with m^3 and h , but the K^+ channel is multiplied with n^4 . Parameter h is an asymptotic parameter acting opposite to m and n . i.e. it decreases while the other two parameters increase, such that h acts as a limiting counterweight against strong increases in $I(t)$.

On below figure 11 one can see the effect of the gating variables on the membrane current (sum of all channels and leakage) changing with input voltage. As it can be seen in the figure, with varying voltage, the membrane current changes the slope at a -65 mV point.

Finally, the output of the HH neuron depends on the input strength. One can increase the spiking frequency of the HH neuron by applying a stronger input. HH model provides a spiking behaviour where the rising edge of the output is jumping very fast such that the rise creates a discontinuity. Such models are called as type 2 neurons. Alternative to the HH neuron's fast rising spike, a type 1 i.e. easy rise spike neuron is Moris-Lecar neuron model (ML) (Venkov, 2008). On below figure the spiking characteristic of the HH neuron is given.

There are simplified or further modified HH models in the literature. Above summarized one is the very early 2 channel (1 for Na^+ and 1 for K^+) HH model (Hodgkin and Huxley, 1952). In literature there are further improved models with increased or decreased number of channels.

ML model different than the HH neuron, provides a complex and realistic behaviour again with only two main membrane channels, but provides a type 1 impulse behaviour. Similar to HH neuron depending on research on squid giant axon, ML neuron is also depending to biological research on barnacle giant muscle fibre (Moris and Lecar, 1981). Type 1 neuron models' output behaviour i.e. relatively smoother spikes, are more realistic in simulating the human neuron behaviour (Venkov, 2008).

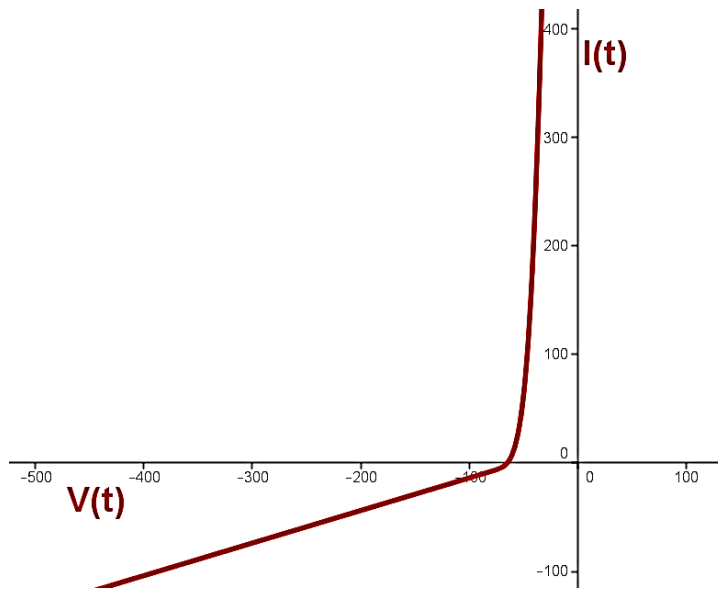


Figure 11. Membrane current versus varying input voltage (mV) representation. -65 mV point is a breaking point where the slope of the I-V line changes. It is named as the resting point. Under 0 input stimuli current from each branch of the dendritic tree, the HH neuron will rest at that point.

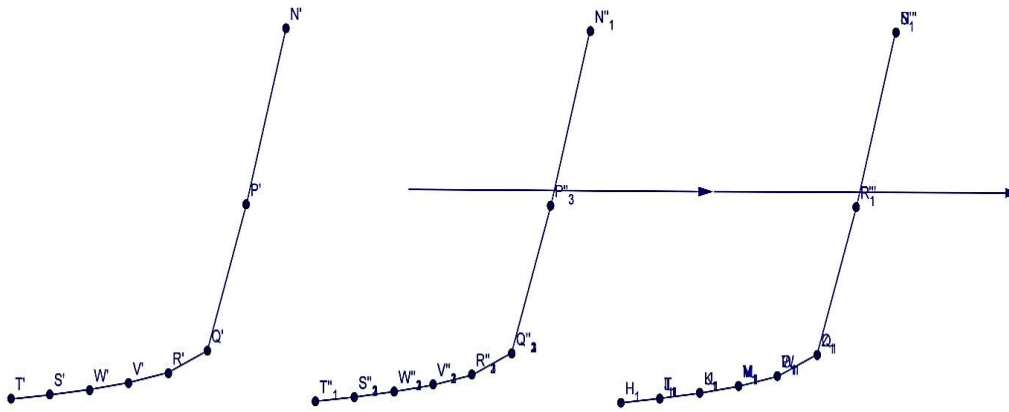


Figure 12. Representation of the HH neuron output spiking behaviour. Horizontal axis is time and vertical axis is the output potential (NOT to scale). The sudden fall of the amplitude of output potential is clearly seen.

The main difference between the mathematical models of HH and ML, is that second one adding a second differential term of K^+ channels (dw/dt). The second ion channel of ML neuron is a Ca^{2+} channel. This second channel is assumed being fast spiking i.e. the sensitive channel. On below equations a summary of ML neuron is given.

$$\frac{dV}{dt} = I - g_{Ca} m_{\infty} (V - V_{Na}) - g_K w (V - V_K) - g_L (V - V_L) \quad (\text{Equation 12})$$

$$\frac{dw}{dt} = \tau_w(v)(w_\infty(v) - w) \quad (\text{Equation 13})$$

Notice in the above model, there are gating variables (less than the HH model), reverse potentials (same as HH model) and time constant functions (only a single one). Below is a summary of those variables.

$$\tau_w(v) = \emptyset \cos \frac{V - V_1}{2V_2} \quad (\text{Equation 14})$$

$$w_\infty(v) = 0.5 \left(1 + \tanh \frac{V - V_1}{V_2} \right) \quad (\text{Equation 15})$$

$$m_\infty(v) = 0.5 \left(1 + \tanh \frac{V - V_3}{V_4} \right) \quad (\text{Equation 16})$$

V_1, V_2, V_3 and V_4 are constant values decided for ML neuron. Similarly reverse potentials are known values.

If one compares the ML neuron with the HH neuron, a very fundamental difference is observed easily. The HH neuron contains a limiting gating variable h , which does not exist in the ML neuron. The limiting variable h of HH neuron keeps the model output increasing behaviour within a predefined rate. The ML neuron is relatively simpler and does not contain a limiting variable and all behaviour of the model is controlled with conductances, reverse potentials and two simple gating variables.

HH, ML or any other conductance based neuron models aim to implement and simulate all neuron behaviour in a very realistic approach. In such case, the models can go out to very complicated forms. When the purpose is to construct a bio-inspired artificial intelligence or making a model of cortical activity (on level of population dynamics or network), networks built up with such neurons can be very difficult to solve numerically. When a network model is to be constructed, a simpler model may be needed. The next section gives information a relatively simpler type of model which is easier to use with networks.

3.2.2 Spiking Neurons Modelling Approach

As mentioned at the end of section above, when a simplified neuron model with acceptable output quality is needed, spiking neuron models are preferred in spite of complicated conductance based neuron models. A spiking neuron model is a neuron model which generates a spiking behaviour (i.e. output activity) without using the biologically motivated ionic conductances. In other words in a spiking neuron model, the conductance based spiking behaviour is approximated with a mathematical equation which is not using biological channelling models. It is clear that the conductance based neuron models are also producing spiking behaviour but notice the “spiking neuron” name is used for relatively simpler models.

The two very famous spiking neuron models are the integrate-and-fire neuron (INF) and the leaky integrate-and-fire neuron (LIF) both of which use the same capacitance type equation given below. The major difference between the INF and LIF is that the leakage function $F(V)$ is linear for LIF i.e. each LIF is an INF but with simplified $F(V)$. The mentioned function $F(V)$ replaces the $I_k(t)$ of the conductance based models so it holds

the place for all ionic and leakage channels (Gerstner and Kistler, 2002; Jolivet, Lewis and Gerstner, 2004; Venkov, 2008).

$$C \frac{dV}{dt} = -F(V) + I(t) \quad (\text{Equation 17})$$

On network usage of LIF or INF, it is expected that the model provides a spiking behaviour between a resetting voltage and a spike threshold. With other words, a INF or LIF neuron output starting from the resting potential, starts to rise with the aid of external stimuli, up to the spike threshold, then starts to decay very rapidly down to a resetting voltage below resting potential. This oscillation between two values is the spiking behaviour of the spiking neuron.

By integrating the equation 17 under a constant input I_0 ; one obtains a very simple form of output given in 18.

$$V(t) = I_0 + (V_0 - I_0)e^{-\frac{(t-T_m)}{C}} \quad (\text{Equation 18})$$

In the above equation, the term T_m is the spiking time, C is a constant, I_0 is the constant input and finally the V_0 is the initial condition of the $V(t)$. Computed $V(t)$ value reaches to the I_0 , if the t increases to infinity after T_m , so we can tell that the I_0 is an upper limit for the $V(t)$. Therefore in cases where the input I_0 is below the h (spiking threshold), there cannot be any spiking behaviour. If the I_0 is above h , i.e. an input strong enough to penetrate the system exists, then spiking behaviour arises with a time interval between the spikes as given on below equation. The $V(t)$ output increases up to h (since I_0 is the upper limit above the h), then falls down to V_0 and starts to rise again. The increasing duration up to next decrease is as given.

$$T = C \ln\left(\frac{I_0 - V_0}{I_0 - h}\right) \quad (\text{Equation 19})$$

On below figure 13 the shape of the output for a very simple LIF model simulation is given.

For further information on LIF and INF models and networks please refer to Bresloff and Coombes (2000). LIF/INF neuron models are the most common neural models used on inverse problems and development. There are studies where the LIF output matches with biologically based models with very high coincidence rates (Brette and Gerstner, 2005). While unifying such neuron models (HH, ML or LIF/INF) one can add synaptic response embedded into the neuron model or can externally implement synaptic models between neuron models. LIF models do not include the synaptic model internally so when unifying two LIF models one needs a synaptic model in between.

As it can be seen in the above models, a neuron model is expected to produce a voltage type output activity value. So the input to a synaptic model will be the neuron's postsynaptic voltage. The synaptic "gap" is assumed to hold a voltage and transform it into a dendritic current.

So a current equation based on the synaptic conductance can be easily written in the form of below equation:

$$I_s(t) = g_s s (V_s - V) \quad (\text{Equation 20})$$

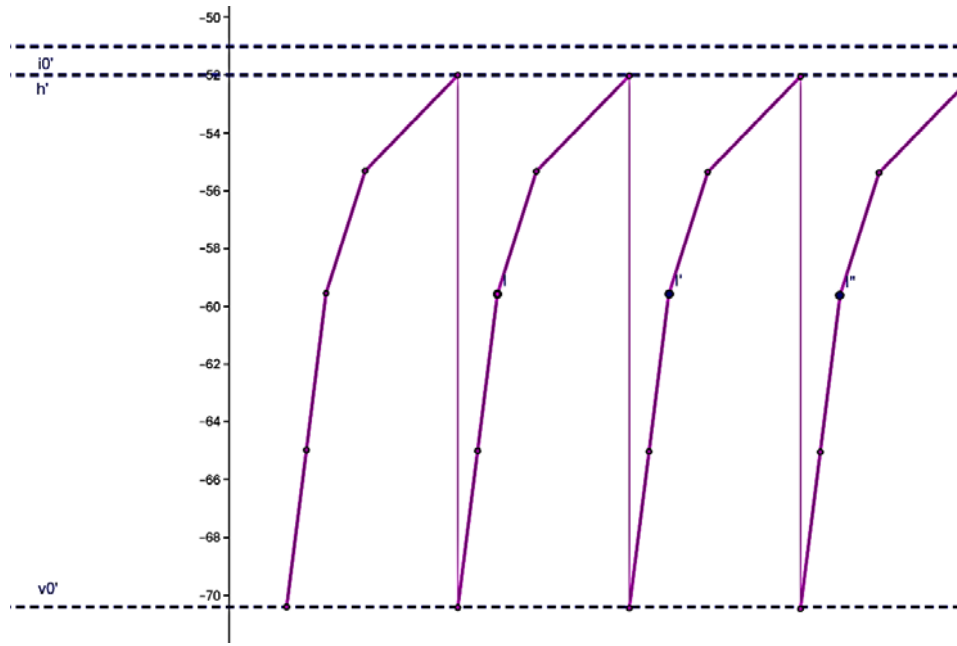


Figure 13. The representation of the output for a Leaky Integrate-and-Fire (LIF) neuron model. Horizontal axis is time in ms. Vertical axis is output voltage in mV.

Above equation represents the synaptic current $I_s(t)$, where g_s is the synaptic conductance, s is the gating variable, V is the input voltage and V_s is the reversal voltage. The most common approach in approximation of the postsynaptic voltage or current is to use a predefined transfer response shape. Below is two common forms of post synaptic shape which can be used to approximate the value of $I_s(t)$.

$$P(t) = \alpha e^{-\alpha t} H(t) \quad (\text{Equation 21})$$

$$P(t) = \left(\frac{1}{\alpha} - \frac{1}{\beta}\right)^{-1} [e^{-\alpha t} - e^{-\beta t}] H(t) \quad (\text{Equation 22})$$

Both of the above functions are shapes for postsynaptic response. The first one is an exponential decay with a Heaviside function where α is the decaying rate with time. The second one supplies two different exponential parameters α and β , which can be used to adjust the rise and fall times of the response. Since this thesis is far more focused on rate models and neural population dynamics –especially the DFT- further details of synaptic, dendritic or neuron models will not be given here. On below equation 23, the structure of the $H(t)$ is given which is used in above equations 21 and 22.

$$H(u(x', t)) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \quad (\text{Equation 23})$$

3.2.3 Networks of Neuron Models to Construct Intelligence

Current network models containing spiking neuron models depend on processing of a representation (such as x) inside the network so that each functional part of the cerebral cortex is individually modelled to perform the intended function (such as $f(x)$). As much as the functionality increase, total number of used nuclei or cortex increase as expected and as long as the function is complex the required number of neurons is also expected to

increase. Although The neuron models used in previous paragraphs contain generally non-linear transfer functions with various defining parameters and inputs (except the LIF). A highly complex network of such non-linear nodes can be very difficult to numerically solve. In such cases transformations (from non-linear to linear or less non-linear) are used. This application is named by Eliasmith and Anderson (2003) as neural transformation. One important point is that the encoded representation x is not decoded but the resulting output of the $f(x)$ is decoded. Thanks to their biological base, such models may be task independent different than the classical AI. Additionally with such models, very dynamic behaviour can be achieved since the neural network allows implementing of many different type of transformations, functions, peripheral and central encoders/decoders. The most limiting (or maybe only) input of such an architecture is the limit of the hardware processing dependence to time. Some highly complex networks may reach to computational complexity levels which can never be achieved in earthly limited time.

Eliasmith provides experimental results of SPAUN which is a complex brain model which achieves success in various tasks (Eliasmith, Stewart, Choo, Bekolay, DeWolf, Tang and Rasmussen, 2012; Choo and Eliasmith, 2014). Notice the task independence of such architecture makes this approach very different than classical AI which is most of the time focused on specific tasks like playing chess or recognizing a face. Bekolay, Bergstra, Hunsberger, DeWolf, Stewart, Rasmussen, Choo, Voelker and Eliasmith (2014) provide an open source framework to build such models namely NENGO. Hurzook, Trujillo and Eliasmith (2013) reports successful iterations of visual and perceptual tasks via complex neural models using NENGO. Stewart and Eliasmith (2011) use a network of LIF models to perform a Tower of Hanoi task.

The network approach to construct bio inspired models of intelligence is a very novel and promising area with known limits of computational complexity. Despite Gödel's claim against the impossibility of modelling "human scale" intelligence (Aaronson, 2011), the study of bio-inspired network modelling continues to provide very successful results. Aaronson (2011) reports a proof by Kurt Gödel showing the impossibility of the successful and complete modelling of human brain with present computers. Although the numbers of 100 billion of neurons are not reached in existing network models, Eliasmith and colleagues have published very impressive results. One example is the SPAUN, a 2.5 million neuron network including capabilities like working memory, sensory input processing and motor output decisions (Eliasmith et al. 2012; Choo and Eliasmith, 2014). For interested reader, it shall be mentioned here that an open source software framework to build such networks is available online for interested readers supplied by a team located at University of Waterloo (Bekolay et al, 2014).

3.3 Rate Models

In contrast to the models of networks of neurons or single neurons, the population dynamics approach focuses on modelling the highway rather than modelling many cars on that highway. This is the main advantage of the rate models.

Notice that rate models provide simplicity on mathematics but there are other benefits as well. A mathematical model of a neuron running in a computer is deterministic by its nature. As expected the network using such neurons is also deterministic. One can add noise and probabilistic patterns to form a stochastic network with stochastic patterns and later make his/her analysis using multiple trials in such stochastic networks, but notice originally individual neuron model with spiking characteristics shall be discriminated

with its inherent/innate stochastic behaviour from a rate model with added noisy behaviour. Rate models are advantageous due to their already averaged behaviour. In other words, rate models intrinsically contains stochastic behaviour averaged. The only gap of a rate model is that it lacks arising of odd behaviour due to spiking characteristic of a single neuron (Eliasmith 2013).

Cuijpers and Erlhagen (2008) provide a detailed description of the development of a rate model from a simple neuron model. Below description of the fundamentals of rate models are based on the manuscript by Cuijpers and Erlhagen (2008). After summarizing the connection between the rate model and the spiking neuron, we will move to our major field of study, the dynamic field theory.

In previous sections a brief summary of the LIF neuron is given. An existing model of averaging LIF model behaviour is called as the ‘‘Sigma (Σ) node’’ neuron. Σ node gives a relation between the membrane potential of the LIF neuron and its spiking frequency (firing rate). The very simple relation between the firing rate r and the membrane potential u is given as;

$$r_i(t) = f(u(t)) \quad (\text{Equation 24})$$

In above equation 24 r_i is the time dependent firing rate function, $u(t)$ is the membrane potential (it is used as the internal state of the Σ node) and finally $f(u)$ is a thresholding function. $f(u(t))$ can be in the form of the Heaviside function given in equation 23 or can be a sigmoid function, both forms are popularly used. The sigmoid function will be used very much in later sections of this thesis. On below figure the representation of a Σ node is given. As it can be seen various inputs can be connected to a node and Σ means a linear superposition of such inputs. At the end what is achieved is a summing equation for the rate of membrane potential changes of the node. See equation 25 below please.

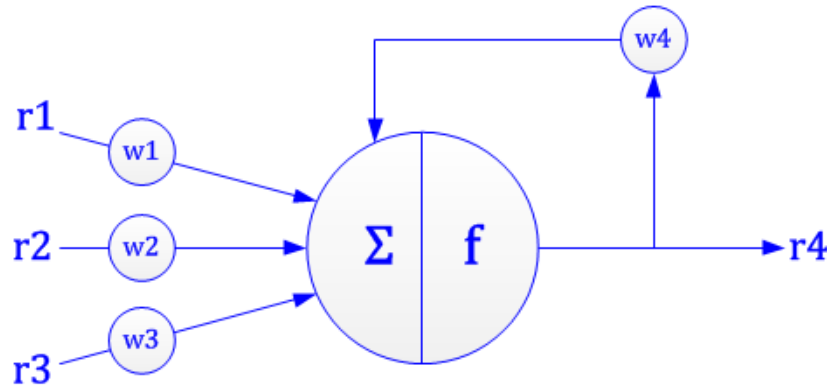


Figure 14. The sigma node representation. Original drawing including same theme in Cuijpers and Erlhagen (2008) slightly modified. Please refer to mentioned reference for further information. Notice the similarity between the neuron schematic given in figure 3 and above Σ node.

$$\tau \frac{du_i(t)}{dt} = h - u_i(t) + \sum w_{ij} f(u_j(t)) \quad (\text{Equation 25})$$

The membrane potential changes in a dynamical way (refer to figure 9 and equation 111 for a definition of ‘‘being dynamical’’ for a system) under control of the interconnection

weight w_{ij} of the synapse between the i^{th} and j^{th} neuron. The constant τ provides a smooth decay of the output since it is the time constant of the system. The system output (membrane potential) increases away from the resting potential h during the activity and decays back to the h while activity is fading.

Above given equation 25 and the Σ node concept is a good point to discuss the dynamic field theory since it is the representation of a simpler idea of Amari equation (1977). The Amari equation and dynamical field theory subjects will be discussed in the following subsections.

3.3.1 Dynamic Neural Field and Dynamical Field Theory

Dynamic Field Theory is an embodied and situated approach to understanding and modelling human cognition which has been fastly developing in the last two decades. Embodiment is a term pointing out that the human cognition is “running” in a very close coupling with the system it is embedded in and the environment which is sensed by the sensory apparatus of the mentioned system (Robbins and Aydede, 2009). In other words, embodied cognition perspective claims that the neuronal processes which are the “basis” for the behaviour of the organism cannot be abstracted from the environment and the embodying system tissues’ associated sensory and motor processes (Schöner, 2007). In Schöner’s words, cognitive processing is never off-line (2008, p.117). Thus according to that approach, any claim on modelling a cognitive process using any sensory input and producing motor output shall by its nature consider the nature of the continuum between the sensor-process-output-environment-sensor loop to succeed in simulating the behaviour generated by an embodied cognitive agent. Although there are symbolist arguments against the embodiment approach, the embodied dynamic models have various robotic applications which are very successful on achieving coupling with the environment in a continuum and giving dynamic reactions to the changes of the environment especially for spatial tasks. These robotic applications will be covered on paragraph 3.3.5. Embodied/dynamical cognitive system approach is claimed to be inconsistent with classical modularity of mind theory and dissolving the boundaries of anatomic modules (Anderson, Richardson and Chemero, 2012).

Dynamical system approach posits that the firing rate values of a neuronal system can be assumed as the states of that system, which is called the rate code principle. For example, the present firing rate of a group of sensory cells may be used as the state of the group. Secondly, the firing rate of the same sensory cell group may also represent the spatial properties of the sensor’s observation (Schöner, 2008). This is a space code. Again as an example the metric (spatial) location of an object within the field of view of the sensor may be coded as an activity region in a two dimensional spatial field of view plane. In such a model the centre position of the axis will be defined by the object position within the observation tolerance functioning as the standard deviation of the shape and environmental/systemic noises affecting the activities amplitude and shape. The CEDAR framework used for modelling purposes in this thesis contains both the rate and space code functionalities (Lomp, Zibner, Richter and Schöner, 2013), in later paragraphs about CEDAR these properties will be clarified with further explanations. By accepting the rate and space code approach, the field as a dynamic system covers the sensor-cognition linkage far better than the connectionist approaches, since the peripheral devices of the cognitive apparatus such as sensors and muscles are linked without eliminating the properties like noise, observation tolerance, sensory capabilities, reaction times, and those are all embodied and considered (Schöner, 2008). Oppositely; many neural network

models underestimate even the effect of the noise and stochastic contributions on the determination of the system connection weights.

The gap -subject to a long (very long) term debate- which is between the representation and the body (the lost/undiscovered gap between syntax and the semantics) is claimed to be filled with a new representational language by DFT theory. General acceptance of modern dynamical approaches is the denial of the existence of the representations in the human cortex (Fodor and Pylyshyn, 1988). Oppositely and surprisingly, although it is a dynamical approach, DFT does not deny the representational language but rather claims that the space and rate codes (i.e. the field activity) form the representation of the mental language (Schöner, 2007, 2008). This results in a cognitive system where the firing activity of a nuclei or cortical area serves as a representation to be manipulated by embodied field connections. Connections between these processing cell populations are used to process the field supplied by the source. In this way DFT provides a connectionist approach further realizable and dynamic. Despite Schöner's critics (2007, 2008) on connectionism as a major researcher of DFT, DFT approach is essentially a connectionist approach where the weights are adjusted via kernels, but provides multiple additional positive properties such as above mentioned embodiment and dynamic behaviour.

Neural field studies starting from late 1950s takes the firing rates as the primary state variable by assuming the populations of neurons embedded in "coarse-grained areas" (Coombes, 2012). Beurle, Wilson, Cowan and Amari created a basis for neural field modelling via their construction of mathematical foundations. Starting from Beurle's (1956) work focused on understanding large scale neuron population excitation behaviour, adding the inhibitory capability with Wilson and Cowan (1972), and ending with Amari's (1977) foundation of Mexican hat type kernels, a mathematical model for neural population activity is at hand.

Briefly summarizing, Integrate and Fire Neuron (leaky or not) approach simplified single cell modelling to provide an ease of population modelling. But DFT goes further ahead of that approach by directly modelling the overall population dynamics by omitting the single cell models and spiking behaviour. Notice this approach forms for the researcher a biologically based (at least on population dynamic layer) modelling platform.

While speaking on the neural field modelling studies, current researches may be organized under two main groups. Most of the time dynamical neural field (DNF) name is used for such studies. DNF studies may vary in a very large scale group including, studies focusing on creating further realistic and accurate neural field models, studies aiming to understand the mathematical behaviour of the very complex neural field equations (some portions remain still numerically unsolvable) and studies using the DNF in robotic applications. Although in fact robotic studies is clearly a part of DNF area, due to the large amount of the work focused on the subject, one may take these portions as the second area. Although there are rare cases the DNF name is used, most of the time the preferred name for such robotic studies is DFT. So clearly DFT is a sub area of DNF but the amount of the DFT work is quite high both in modelling and robotics; so that most of the time, each DNF approach based on Amari equation can be accounted under the column of the DFT studies (Bicho, Louro and Erlhagen, 2010; Venkov, 2008).

There are multiple application areas for the DNFs besides DFT. One of the relevant areas under the DNF flag is the Dynamic Causal Modelling (DCM) area which is the study of comparison and estimation of biophysically plausible models using neural imaging or encephalogram data. Simply DCM uses models to estimate neural experimental data and

the experimental data to obtain the model parameters. The second part (finding the model parameters from experimental measurement) is namely the inverse problem, and is also used mathematically to decide the kernels of population dynamics in DFT applications (Pinotsis, Moran and Friston, 2012; Potthast and Graben, 2009). One further example of usage of DNF is to shape similar behaviour to observed neural behaviour using DNF models. Such example may be seen in Satel, Story, Hilchey, Wang. and Klein (2013) where fields are used to understand cognitive phenomena or inhibition of return.

Before starting the DFT subject, an introduction to the dynamics of system concept may be useful. A system is named as dynamic if the current rate of change of the system depends on the current state of the system. So in a dynamical system, the current state defines how the next state will be achieved and what it will be, i.e. how fast and where. Therefore the rate of change of a dynamical system shall be function of the input parameter of that system as shown in below equation 111. Even there are various differing definitions of “dynamic system” regarding the study area, for neural modelling; one brief and good definition can be given as follows: A system is dynamic if the rate of change is one of the state variables (Van Gelder, 1998, p.618).

$$\frac{dx}{dt} = f(x) \quad \text{(Equation 26)}$$

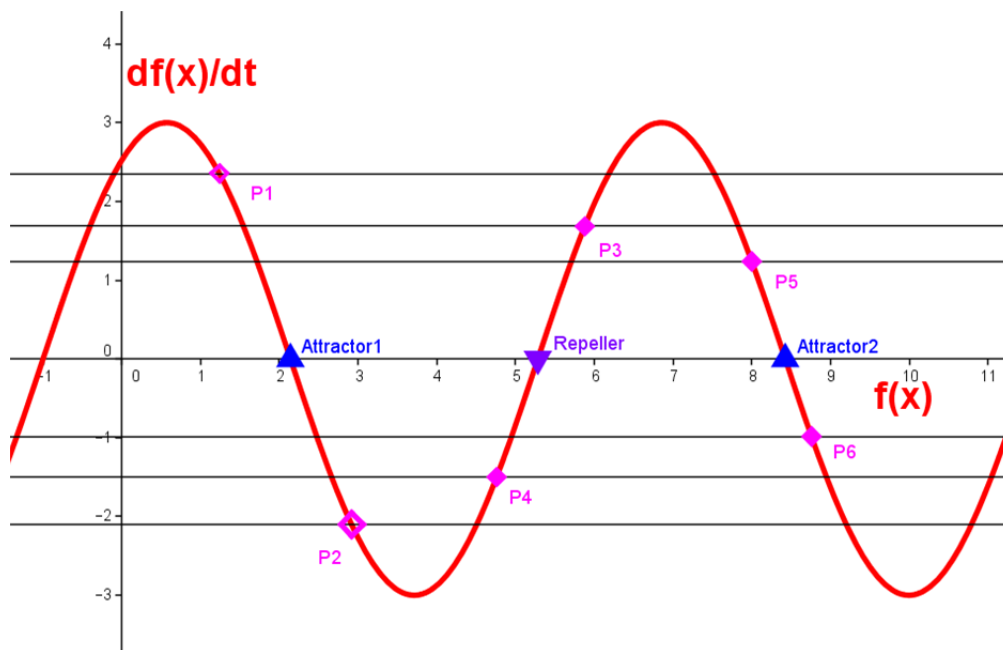


Figure 15. A dynamic system and states representation. At point P1, the slope of the system (i.e. dx/dt) is positive. So the system output (i.e. the value of $f(x)$) will increase until the system rate of change is equal to 0. This is the point which is called as Attractor1. At point P2, the system has a negative rate of change (i.e. negative first derivative). So the system output will decrease and again move towards point Attractor1. The same situation happens at point P4 also. Notice the rate of change is equal to 0 at point Repeller, but still the system moves towards Attractor1 on left of the Repeller even in very close proximity, due to the repelling behaviour of that point. On right side of the Repeller the Attractor2 performs as an attractor for all points such as P3, P4 and P5. This

figure clearly shows why the attractor states are the stable states attracting the system behaviour to rest upon them.

As seen on equation 26, each value of x will create a new rate of change on x . simply if there is only one point where the rate of change of x is equal to 0, this means that at that point x will no more change. Such points are called as attractors since if currently x is changing, it will keep on moving up to that point and then stop changing at that point since the derivative is equal to 0. DFT assumes that human neural activity contains such stability states where the system spatial variable stops changing unless any perturbation to move out from the attractor is applied to the system.

The characteristics of the system may be changed with emerging system architecture or nature, but at the end, the system will relax to a new attractor state so that the stability will be obtained. Therefore the stability is one important feature of the dynamical systems. Please see figure 15 above.

There can be some cases where a zero crossing point forms a depletory opposite to the attractor concept where the depletory is squeezed between two attractors. In such case the depletor point is located in such a manner that it is in the zero crossing point of $x'-x$ graph such that the x' value just before the depletory are negative and just farther from that point are positive. In such case the negative x' cause the function value move away from the depletor until it reaches to an attractor i.e. stable state with a varying change ratio due to the non-linear behaviour of $x'-x$ curve. This case is represented in figure 15 above.

The DFT approach assumes that there exist stable states in the dynamical behaviour of a cortical plane, nuclei or column. Such states can be obtained by solving a mathematical model of the subject neural population. While speaking about the firing rate of a neural population, Amari equation provides a model where the local stable states are solvable due to the easy form of the Amari equation in steady state. This issue will be further clarified in later paragraphs.

When constructing a dynamical model of the neural population activity, the inactive cases shall also be considered. In DFT approach such inactivity situation is founded with negative resting level which is necessarily inhibitory. Such inhibition provides that the population activity is suppressed by the negativity of the resting level in lack of input. When there is an input, its strength, the self excitory behaviour of inside the population and the population's connections to neighbours shall all be enough to crossover a threshold to find an active field. So the model of course shall include an input. Additionally there might be an expression describing the inter-population and intra-population dynamics. In the DFT approach such connectivity function is named as Kernel. Finally the overall model shall be in the form of a dynamical equation i.e. it shall contain the relation in between the field activity and the field activity's rate of change. The Amari model contains all the above mentioned properties, therefore, additional to all experimental data on behalf of the neural field approach, it also contains an intuitively suitable structure to model the population dynamics of a neural entity. Below is the Amari equation (Amari, 1977).

$$\tau \frac{du}{dt} = -u(x, t) + h + S(x, t) + w(\Delta x) \circ f(u(x', t)) \quad (\text{Equation 27})$$

$$\tau \frac{du}{dt} = -u(x, t) + h + S(x, t) + \int dx' w(x - x') f(u(x', t)) \quad (\text{Equation 28})$$

$$\tau \frac{du}{dt} = -u(x, t) + S(x, t) + h \quad (\text{Equation 29})$$

The above equation 29 represents an imaginary condition where neural populations are completely independent of each other i.e. there is no connection pointing out a lack of global neural network, where the equation is completely simplified since the Amari equation takes a form where the system dynamic depends only internal behaviour and input (Bicho, 2000).

In above equations the term $w(\Delta x)$ term is the interaction kernel and is convolved with a threshold function $f(u(x', t))$ which is used to suppress the kernel's excitory parts in case of the system dynamics are below a preselected threshold. The terms are listed below:

τ : Time constant for the population dynamics. This term defines how fast the activity patterns rise or decay.

$u(x, t)$: the field activity function. The output of the U function provides the activity value.

h : resting level of the model. This is the point where the system rests when there is no input. It is used to prevent the system being active without getting an input.

$S(x, t)$: the input(s). This input can be in form of any arbitrary signal but if this is the resulting output of any sensory process, it is rather preferred being Gaussian for containing the sensory tolerances and uncertainties. S function may be a single function or in form of sums of various inputs from varying sources such as sensors or memory cells.

$w(\Delta x) = w(x - x')$: Interaction kernel representing each neurons excitory or inhibitory relation from the neuron at the origin with a distance $x - x'$.

$f(u(x', t))$: The threshold function. This function defines a threshold crossing expression for any neuronal activity at distance $x - x'$ from the origin.

$w(\Delta x) \circ f(u(x', t)) = \int dx' w(x - x') f(u(x', t))$: Convolution of the kernel and the threshold over the defined spatial scale. This convolution is defined over all the space that the spatial coverage region. In case of the sensor field spatial value is an angle such as head rotation angle, this can be limited between 0-360° (or 0 to 2π for radian computations) in spite of infinite space. In other words limits of Amari convolution can be adjusted to define a spatial coverage for the corresponding neural population to provide a mapping between the spatial parameter observation and the population activity. This is a way of representing the outer world in the form of embodied neural activity.

The threshold function $f(u)$ can be selected in many forms but the most popular and realistic behaviour is obtained with a sigmoidal shape. Below are also the equations defining different kernel examples (equations 30, 31, 32 and 33).

$$w(x) = A e^{\frac{-(x-x')^2}{2\sigma_1^2}} - w_{inh} \quad (\text{Equation 30})$$

The equation 30 provides a relation of neural interaction between neural fields depending on the distal separation between them i.e. it is a function of a spatial parameter. Notice the

activity relation here is simply independent of time but it is a function of the separation between fields' physical distances and forms a Gaussian including a standard deviation σ . Now let's look at the terms of the eq. 30.

A : The excitation constant which will vary with a Gaussian behaviour over distances between fields. This term defines the maximum allowed level of excitation. As the A value increases, the excitory effect of a field in a neighbouring one is also potentially increased. Since it is here representing the maximum value of a excitory field, it can also be named as w_{exc} .

w_{inh} : A constant value of inhibitory effect between neural fields. Notice excitory or inhibitory w (i.e. $w_{exc}(x)$ or $w_{inh}(x)$) does not need to be constant or varying. For creating various behavioural dynamics and creating different networks, one can play with the shape of interaction kernels. In below example equation 31 and 32 interaction kernel with both Gaussian excitory and inhibitory interactions and additional forms are given. On figure 16 the shape of some Gaussian kernels are given with varying inhibitory and excitory parameters.

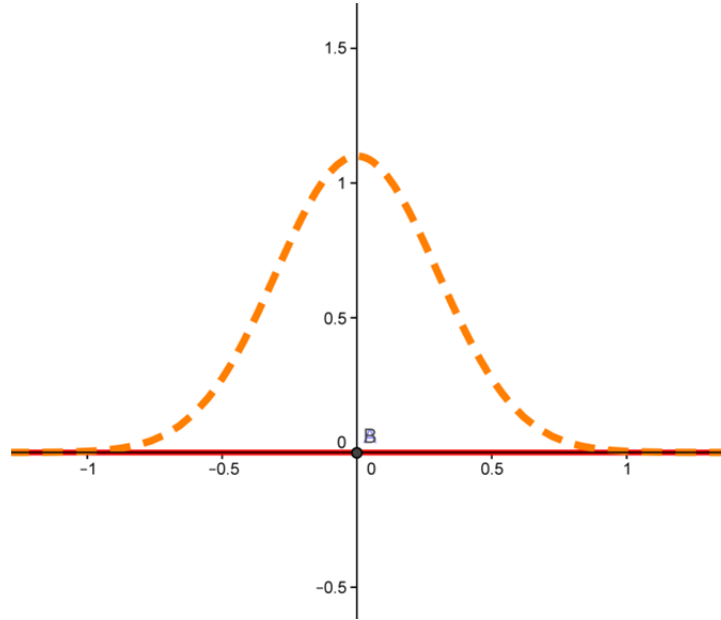


Figure 16. The representation of an interaction kernel in form of a Gaussian with 0 amplitude global inhibition. Notice in all around the x axis (the spatiotemporal axis), the kernel never falls below 0. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter

$$w_1(x) = w_{exc} e^{-\frac{(x-x')^2}{2\sigma_1^2}} - w_{inh} e^{-\frac{(x-x')^2}{2\sigma_1^2}} \quad (\text{Equation 31})$$

$$w_2(x) = w_{exc} e^{-\frac{(x-x')^2}{2\sigma_1^2}} - w_{inh} e^{-\frac{(x-x')^2}{2\sigma_1^2}} - w_{inh2} \quad (\text{Equation 32})$$

$$f(u(x', t)) = \frac{1}{1 + e^{-\beta(u(x', t) - u_0)}} \quad (\text{Equation 33})$$

As it can be seen with the above described kernel shape flexibility, one can define various interactions and multilayer fields to provide different cortical behaviours. But notice the form given in equation 30 is a proper one which allows creation of a well-known kernel shape since it provides excitory behaviour within the variance (σ^2) region and constant inhibition beyond that point. Such behaviour is very acceptable for a neural field due to the negligible effects of infinitely distant neural fields, so it is expected that only close fields may create excitory behaviour and far fields shall not contribute the processing.

Of course in the above equation 30, the threshold can be replaced with various other functions in case of simplicity required. One very simple threshold function can be a step function $H(u)$. A step function provides an output different then 0 when the threshold is crossed over, but provides inactivity below the threshold, i.e. different then equation 30, does not activate with a non-linear increasing behaviour but provides discontinuity on the rising edge of the activity. A discontinuity due to the impulsive behaviour at threshold or simply at the rising edge can be mathematically problematic so in many situation best again a simple approach may be to use a simplified version of the step function in form of below equation 35 where the discontinuity is replaced with a modified version of the classical step function. The equation 35 is a modification of equation 34.

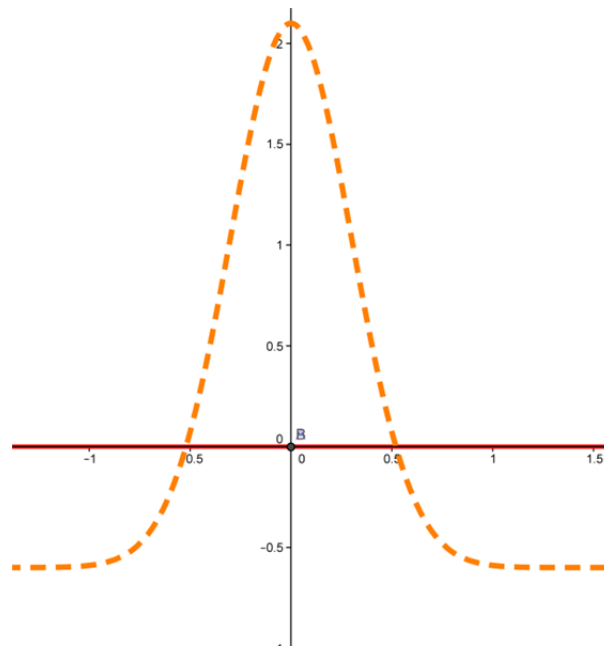


Figure 17. The representation of an interaction kernel with negative and constant inhibitory global behaviour. Notice there is only an excitation near the origin and neighbourhood. After a distance of σ in each direction, a constant negative inhibition prevents the kernel from being positive. Different than the above 2 kernels in figures, there are wizard hat and Mexican hat type kernels. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter

$$H(u(x', t)) = \begin{cases} K * 0, & u(x', t) < 0 \\ \text{impulsive}, & u(x', t) = 0 \\ K * 1, & u(x', t) > 0 \end{cases} \quad (\text{Equation 34})$$

$$H(u(x', t)) = \begin{cases} K * 0, & u(x', t) < 0 \\ K * 1, & u(x', t) \geq 0 \end{cases} \quad (\text{Equation 35})$$

K in above equation is a gain which can be used to provide further excitory behaviour of the system. But this is a method out of classical DFT approach. If one prefers to handle the interaction completely with the kernel K shall be taken as equal to 1. In this thesis, all threshold functions are used as sigmoid and step functions are not included. Below figure represents a sigmoid shaped threshold function.

Excitory and inhibitory kernel usage on formation and shaping of neural field behaviour is a common method on dynamic neural filed studies, not only in DFT but also in relative study areas. A very efficient usage of such methods (namely Jansen and Rit model) on dynamic causal modelling can be seen on Pinotsis, Moran and Friston (2012) where three different neural populations are used on modelling a dynamical behaviour with data earned from encephalographic methods. One of these three populations is inhibitory and two other are excitory resulting in realistic field behaviour.

In a previous chapter a brief introduction to human neuron is given. In a human neuron 10 to 40 % of synapses are known to be inhibitory and those are mostly the ones closer to the soma to provide inhibition from distal fields (Venkov, 2008). One very well-known example of usage of excitory and inhibitory behaviour together can be seen on the ganglion and bipolar cells of mammal eye (Palmer, 1999). Naturally any retinal tissue of the eye sensor is transmitting the “acquired” signal via a ganglion cell. Ganglion cells can be connected to the bipolar cells directly or via horizontal cells. In most of the cases a horizontal cell is connected to many peripheral ganglion cells while the spatially central ganglions (such as the ones at fovea) are directly connected to bipolar cells. In any way, if a ganglion cell directly connected to the bipolar cell is excitory, the indirect connection of the peripheral ganglion cells to the bipolar cells over the horizontal cell is inhibitory, in other words the direct connection of the central ganglion cells neural network connectivity (kernel) is always oppositely shaped in comparison with the peripheral indirect (using horizontal cells) connectivity.

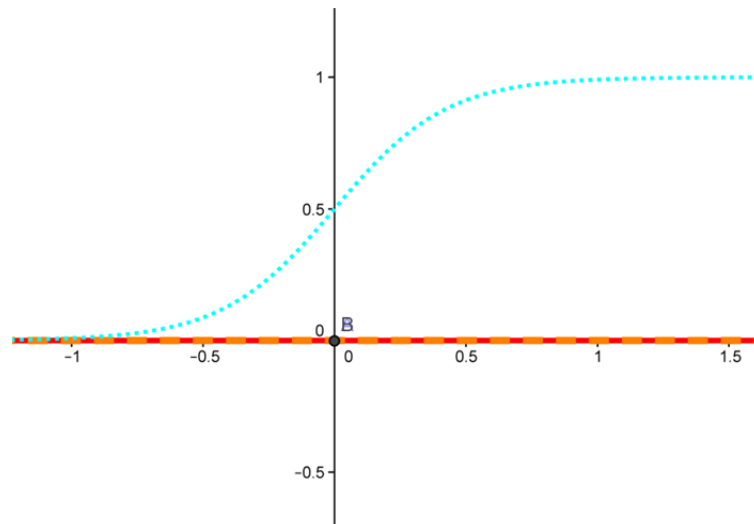


Figure 18. A threshold function in form of a sigmoid. The constant β in equation 30 defines the slope of the increasing region of the threshold. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter

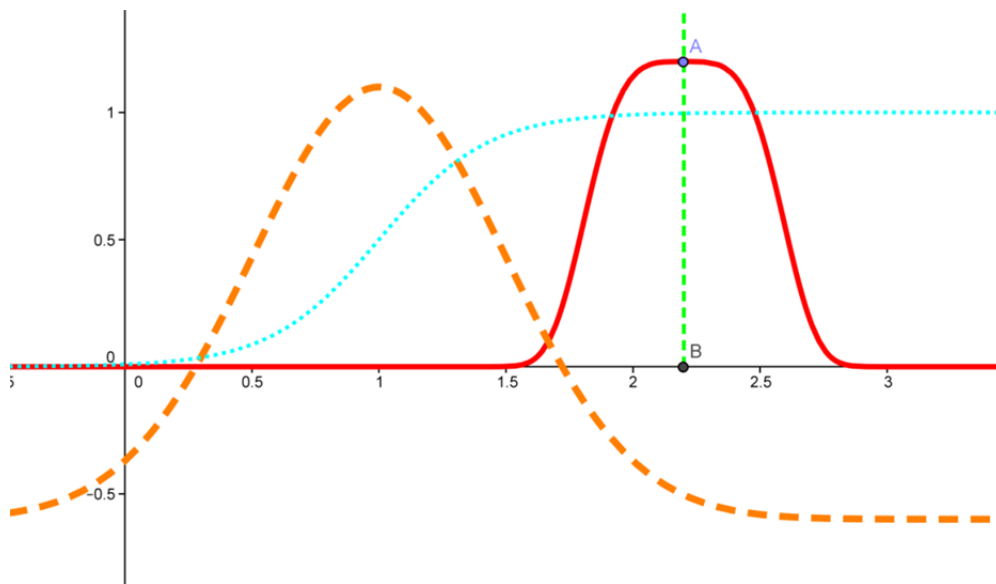


Figure 19. A Gaussian input (red solid line) with mean at point B, and amplitude A can be used as an input to Amari equation ($S(x, t)$). The interaction kernel (amber dash line) and the threshold function (blue dot line) is used to analyse how much and how the function S will be transmitted to neighbouring neural fields and how it will excite/inhibit in the local field. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter

In such case, bipolar cells can easily be assumed like a mathematical summing device which performs an addition operation with excitory central behaviour and inhibitory distal behaviour. At the end what is at hand is a Mexican hat output. For a Mexican hat type kernel please see figure below. In DFT area there are various studies of visual cortex modelling such as saliency extraction, hue and colour determination, position estimation or robotic sensor integration (Johnson, Spencer and Schöner, 2007; Erlhagen, 2002). A DFT/DNF model of cat visual cortex with interaction type kernels can be found at Jancke, Erlhagen, Dinse, Akhavan, Giese, Steinhage and Schöner (1999).

Kernel shape changes with the superposition the sub parts forming a kernel. The one shown in the above figure is in the form of $w_{exc} + w_{inh}$, so it is called a Mexican hat. Adding a constant term to such superposition, one can obtain a wizard hat. In case of excitory terms are greater than the inhibitory terms, those shapes are named as “inverted” Mexican or wizard hats.

Notice all of the above examples contain one dimensional (1D) neural fields, although there are various studies with multi-dimensional DFT models. When the task is to be performed on a 1D spatial environment (imagine angular decisions for turning to a direction) a one dimensional field may be used. In case of planar decision is required such as deciding the location of an object in a plane surface, a 2D model can be used. So; one can increase the dimension of the model or the fields which are used in the model regarding the spatiotemporal requirements.

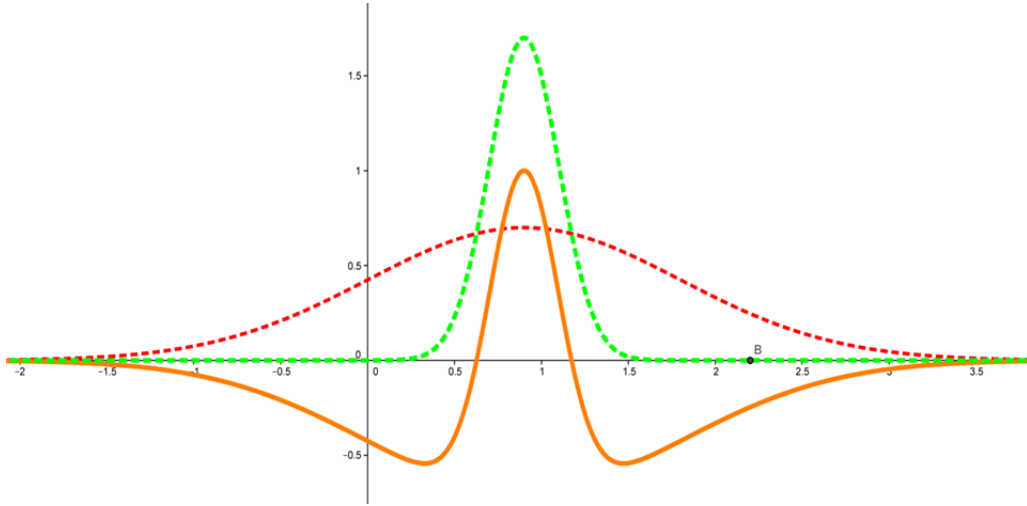


Figure 20. The representation of the formation of a kernel via the superposition of inhibitory and excitory interactions. The green line shows a kernel with completely excitory behaviour (w_{exc}). Red line shows a kernel with complete inhibitory behaviour (w_{inh}). Notice the inhibition characteristics increasing at the mean of the spatiotemporal field. Finally the amber line shows the resulting (superposition) hat type kernel ($w_{sum} = w_{exc} + (-w_{inh})$). The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter

Additionally via using two layer fields, it is possible to create different behaviours. A 2 layer field (2L) can be constructed using one inhibitory and one excitory field unified together. One common method of using 2L method is the Wilson-Cowan method where two populations are modelled to obtain a single neural behaviour. Below is a short summary of Wilson-Cowan model of two populations.

$$\frac{1}{\alpha_e} \frac{\partial u_e}{\partial t} = -u_e(x, t) + \int_{R1}^{R2} dx' w_{ee}(x - x') f_e(u_e(x', t)) - \int_{R1}^{R2} dx' w_{ie}(x - x') f_i(u_i(x', t)) \quad (\text{Equation 36})$$

$$\frac{1}{\alpha_i} \frac{\partial u_i}{\partial t} = -u_i(x, t) + \int_{R1}^{R2} dx' w_{ei}(x - x') f_e(u_e(x', t)) - \int_{R1}^{R2} dx' w_{ii}(x - x') f_i(u_i(x', t)) \quad (\text{Equation 37})$$

Notice in the above 2 population model, both populations are modelled with two kernels; one for defining the interaction inside each of the populations and one for defining interaction between the populations. Such model can be used to obtain complex behaviours which may be difficult with single layer 1D models.

There are various studies aiming to show DFT approach's applicability to general tasks pointing out the DFT as a general language of representation (Simmering and Spencer, 2008; Simmering, Schutte and Spencer, 2007). It shall be noted that DFT is mostly focused on spatial tasks where a spatiotemporal value belonging to the external world is acquired and processed internally by a cortical layer model without a requirement for

biologically basis (Spencer, Simmering, Schutte and Schöner, 2007). But it can be found studies on verbal and non-verbal communication studies (Bicho, Louro and Erlhagen, 2010; Sandamirskaya and Schöner, 2006) or children cognitive development (Spencer et al. 2006; Spencer and Schöner, 2003). Originally the very first studies on DFT is provided from the cognitive development studies area. Cuijpers and Erlhagen (2008) used DFT to model the Bayes rule in a cognitive application. Erlhagen (2003) used DFT to show the possible existence of internal visual models in human cortex. Erlhagen and Jancke (2004) used DFT to model the motion planning capability of human cerebral cortex. On the same subject Bastian, Schöner and Riehle (2003) used *preshape* fields to model motor cortical representations for movement preparation. Due to the very close conceptual proximity between the motor functions and the spatiotemporal variables (motor action is performed inside a “spatiotemporal space”), there are multiple very successful studies on motor function control and motor planning. Also DFT is very widely used to model the planning of movement (Erlhagen and Schöner, 2002). Finally there are some studies where DFT uses some working memory models very often, even in each task (Schutte, Spencer and Schöner, 2003).

Task hierarchy or continuity is an important problem in cognitive modelling. Methods shall be developed for the model to be able to start a new task by perception of the finishing of the previous. For that purpose, discrete movement planning can be performed in form of sequential tasks for reaching continuity in DFT based modelling area (Schutte and Spencer, 2007). Sandamirskaya and Storck (2014) provide a model making multiple DFT tasks sequentially. There are various studies on ordinal dynamics using a “discrete” DFT approach to form hierarchical or sequential task with predefined orders even learning (Kazerounian, Luciw, Richter and Sandamirskaya, 2009; Sandamirskaya, 2013; Sandamirskaya and Schöner, 2010, 2010b, 2010c; Duran, Sandamirskaya and Schöner, 2011; Duran and Sandamirskaya, 2011; Richter, Sandamirskaya and Schöner, 2012; Sandamirskaya, Richter and Schöner, 2011; Sandamirskaya, 2011). In this thesis, we preferred not to use task hierarchy or ordinal dynamics, since the task scheduling subject to this thesis does not work with a first in first out model, but one of the tasks has a priority to interrupt the other. In clearer words, when the pilot observes an obstacle in the flight path, prefers to interrupt the current navigation task and focus on avoiding the obstacle. This is an interruption mechanism, so a method which is different than the ordinal dynamics is used. For details of task sequencing please refer to paragraph 3.3.4.

Parallel to above listed software based models, Sandamirskaya provides an implementation of DFT on a VLSI circuitry to form unified architecture and autonomous learning (2013). This very unique study shall be listed as a hardware DFT effort different than the previous ones.

3.3.2 An Introduction to the Mathematics of Dynamical Systems

DNF area contains many mathematical research and observation to solve such equations. Since the DNF area is much more focused on realistic modelling understanding neural behaviour, equations may be seen in many forms. But in general a single neural field is represented in form of a partial integro-differential equation. If one is speaking of a multilayer neural organisation, the field can easily be represented in form of a state space equation. i.e. multiple Amari (or Wilson-Cowan) equations constructs a state-space. Such systems can be solved by using Fourier transforms to shape a partial differential equation of the system in Fourier domain i.e. simplifying the system by linearizing it. In this chapter, the solution methods of Amari equation will be summarized rather than focusing

on state-space representations. In either case, the neural field equations are quite complex and present literature generally provides methods for steady state conditions (Pinotsis, Moran and Friston, 2012; Venkov, 2008).

Amari equation is in form of a non-linear partial integro-differential equation and is highly complex. In literature there are various numerical methods to solve integrodifferential methods such as VIM (Variational Iterational Method), Local Polynomial Regression (LPR) or ADM (Adomian Decomposition Method) (Batiha, Noorani and Hashim, 2008). None of such methods are easily applicable to Amari equation. Compact finite difference method can be used to solve partial integro-differential equations but this is also not applicable to equations in form of Amari (Soliman, El-Asyed and El-Azab, 2012; Tang, 1993). Numerical methods of modern dynamical systems are generally focused on solving different forms of dynamic equations. Simplifications of Amari equations may be used with simpler methods.

Since Amari equation is in form of an integro-differential equation some simplification may be used for stable state and trapezoidal rules can simply be used. Now let us look at the simplification of the equation for steady state.

On a steady state, the field shall be already reached to an attractor state so that the rate of change of the activity shall be decreased or increased to 0. Please see below equation 38. Notice the τ (time constant) of the equation 39 (and also Amari equation) does not reflect any biological value and it is just a modelling parameter to adjust system behaviour. So cancelling out such term out of the equation does not create any loss on the fidelity. Notice we are focused only to the steady state solution.

$$\tau \frac{du}{dt} = 0 \quad (\text{Equation 38})$$

$$u(x, t) = h + S(x, t) + \int_{x_1}^{x_2} dx' w(x - x') f(u(x', t)) \quad (\text{Equation 39})$$

So in steady state the Amari equation takes the form given in equation 119. Notice in the lack of internal input, i.e. lack of $S(x, t)$, the system shall remain at the attractor state.

So let us define a steady state $\bar{U}(x)$ function which denotes the steady state of $u(x, t)$. i.e. a time independent definition for the attractor state of the Amari equation.

Below equation 40 gives a form of the Amari equation at the attractor state where the system does not vary with time since there is no external stimulus to penetrate the system out of the attractor. If there were such stimulus the system is expected to move out from the current attractor towards a new one.

$$\bar{U}(x) = h + \int_{\check{x}_1}^{\check{x}_2} dx' w(x - x') f(\bar{U}(x')) \quad (\text{Equation 40})$$

Above given steady state integral of the equation 120 is differently limited with equation 39. Notice the full scale $u(x, t)$ is limited in the range (x_1, x_2) which can be an infinite or finite spatial scale. But the \bar{U} term is limited with the range $(\check{x}_1, \check{x}_2)$ where the scale determines the interval where the system is in a steady state interval. i.e. the limits of integral is further squeezed in an interval where u is in steady state.

For simplicity let us use a step function for the threshold function in spite of a sigmoid. In the below function, the only effect of the threshold function is to change the integral limits to a scale where the \bar{U} is above the resting level h .

$$f(\bar{U}(x)) = H(u - h) \quad (\text{Equation 41})$$

In such case the Amari equation is decreased to a very simple form which can be easily solved with numerical or analytical methods. See below equation 42. Notice the interval (x_1, x_2) denotes the range where \bar{U} is above h .

$$\bar{U}(x) = \int_{x_1}^{x_2} dx' w(x - x') \quad (\text{Equation 42})$$

As it can be seen in the above equation 42, the form of the equation at the hand is very simplified. The term $(x - x')$ denotes the distance between fields. So it can be modified to simplify the integral.

$$\bar{U}(x) = \int_{x-x_1}^{x-x_2} dx' w(x')$$
(Equation 43)

Potthast and Graben (2009) attempt to solve inverse problems of Amari equation, i.e. they are constructing kernels from prescribed neural field dynamics. The solution to above integral is simply:

$$\bar{U}(x) = -w_0 x e^{-|x-x_1|} - (-w_0 x e^{-|x-x_2|}) \quad (\text{Equation 44})$$

$$\bar{U}(x) = -w_0 x (e^{-|x-x_1|} - e^{-|x-x_2|}) \quad (\text{Equation 45})$$

The above summarized approach is based on original study by Amari (1977) and for a very detail further analysis of the matter, Venkov (2008) can be referred.

The below two figures give representations of the above equation 45 for a interaction kernel of hat type. The points A and A' are the crossing points of the resting level h and the kernel. Kernel is on the type given in equation 31, 32 and 33 and the threshold is sigmoid. The first figure is the solution where there are no input stimuli and the next one is the steady state solution to the case where there is an input stimuli and a stabilization time has passed. i.e. An input is applied to the system which is in state of figure 21, and the system moved to a new steady state, it is represented in figure 22.

Notice the above solution is valid for single ‘‘bump’’ steady states. There might be cases where there are more than a single bump, i.e multiple activity region arises (multiple attractor states) above the resting level h . Such results may arise from steady state phenomena of both the u function and also from the kernel. Such cases will not be summarized here in detail but the framework used in that thesis to solve DFT models, uses above given steady state simplification to solve steady state activities of constructed models for even in multiple bump case. Details of the framework are given in later paragraphs. Such a multi-bump steady state activity is given in below figure. Notice the unexpected type of the kernel (let us name it cowboy hat) and 3 bumps formed on activity region.

The stability of the solution is an important point on understanding the steady state solutions of Amari equations. One can assume that the system is stable if time dependent perturbations to system decays within time everywhere on the spatial axis. So we can test

such condition with a stimulus to the above solution. Let us assume a steady state stimulus of $\hat{u}(x, t)$ to the steady state solution \bar{U} . Now we have a field activity solution in form of as given in equation 126.

$$u(x, t) = \bar{U}(x) + \hat{u}(x, t) \quad (\text{Equation 46})$$

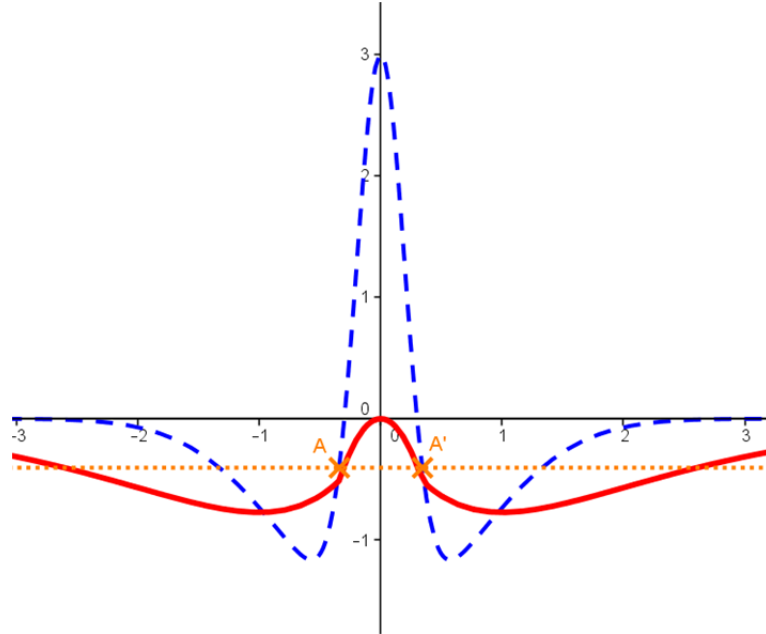


Figure 21. The representation of the steady state solution to Amari equation. Blue dash line is the hat type interaction kernel. Amber dotted line is the negative resting level. Red solid line is the steady state solution. Notice the solution is to a case where there are no input stimuli to the system. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter.

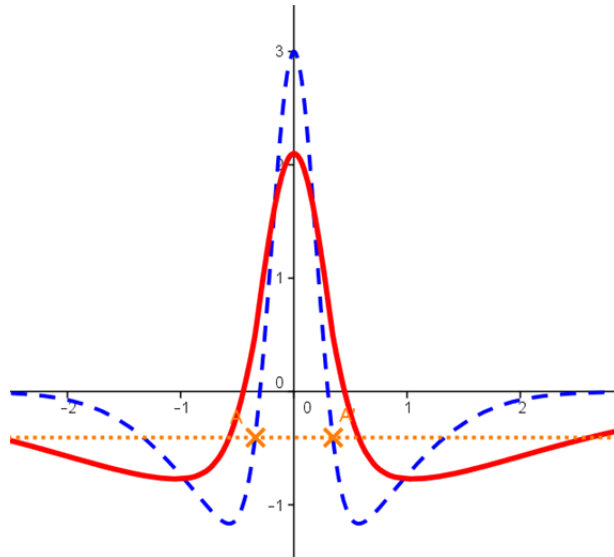


Figure 22. The steady state solution to Amari equation with hat type kernel when there is a Gaussian input to the system (Amplitude of the Gaussian input is 2.1, $\sigma=0.4$). Blue

dash line is the hat type interaction kernel. Amber dotted line is the negative resting level. Red solid line is the steady state solution. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter.

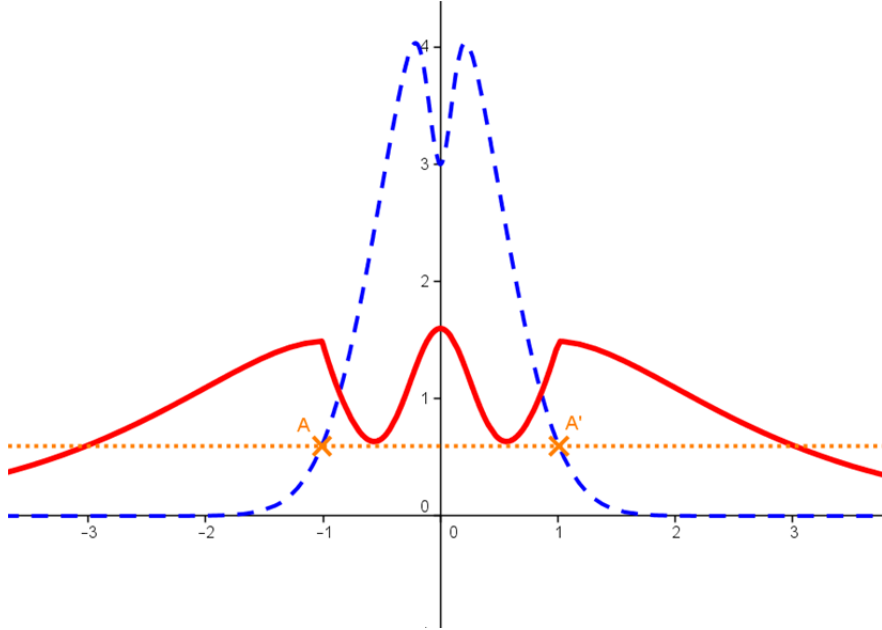


Figure 23. The representation of the multiple bump steady state of a field activity. Blue dash line is the hat type interaction kernel. Amber dotted line is the negative resting level. Red solid line is the steady state solution. The vertical axis is the field activity and the horizontal axis is the spatiotemporal parameter. There is a Gaussian input to the field in this case (Amplitude of the Gaussian input is 1.6, $\sigma=0.4$). Notice even the amplitude of the Gaussian is below the one given in previous figure, still the activity is dominated by the kernel and the positiveness of the resting level ($h=0.6$).

We know that such system will have crossing points on resting level h and we can formulate such points in form of an interval ($x_1 + x_1(t)$, $x_2 + x_2(t)$) where x_1 denotes the steady state and $x_1(t)$ time dependent cases. We will name such points as $x_1(t)$ and $x_2(t)$. We also know at such points the rate of change in time of the field activity shall be 0 in that points as given in equation 47 (Venkov, 2008; Amari 1977).

$$\frac{\partial}{\partial x} u(x_1(t), t) \frac{d}{dt} x_1(t) + \frac{\partial}{\partial t} u(x_1(t), t) = 0 \quad (\text{Equation 47})$$

$$\frac{\partial}{\partial x} u(x_2(t), t) \frac{d}{dt} x_2(t) + \frac{\partial}{\partial t} u(x_2(t), t) = 0 \quad (\text{Equation 48})$$

Using the original steady state simplification of the Amari equation (as given in above paragraphs) we can write the solutions of the steady state at the points $x_1(t)$ and $x_2(t)$ are:

$$\tau \frac{\partial}{\partial t} u(x_i, t) = u(x_i, t) + \int_{x_1(t)}^{x_2(t)} dx' w(x - x') \quad (\text{Equation 49})$$

where $i = 1, 2$

$$\tau \frac{\partial}{\partial t} u(x_i, t) = h - [-w_0 x (e^{-|x_i - x_1|} - e^{-|x_i - x_2|})] \quad (\text{Equation 50})$$

where $i = 1, 2$

Such equation can be simplified as done previously for Amari equation by replacing the time dependent $u(x_i(t), t)$ by a steady state $\bar{U}(x_i)$. With that substitution the equation 50 diminishes to the following form.

$$\tau \frac{\partial}{\partial t} \bar{U}(x_i) x_i = h - [-w_0 x (e^{-|x_i - x_1|} - e^{-|x_i - x_2|})] \quad (\text{Equation 51})$$

where $i = 1, 2$

Equation 51 is an ordinary differential equation and there will be 2 different versions one for $i=1$ and one for the $i=2$ cases. Summing up those 2 equations we have;

$$\text{Let's say } x_1 - x_2 = \Delta \quad (\text{Equation 52})$$

$$\tau \frac{c d\Delta}{2 dt} = h - [-w_0 x (e^{-|x_1 - x_1 + x_2|} - e^{-|x_2 - x_1 + x_2|})] \quad (\text{Equation 53})$$

where $i = 1, 2$

$$\text{Let's say } \tau \frac{c d\Delta}{2 dt} = h - W(\Delta) \quad (\text{Equation 54})$$

In such a system, if the rate of change of the system is positive, then the system cannot be stable. There can be cases where the activity increases or expands to the limits of the spatial axis. So the rate of change shall be negative. 0 rate of change is not expected, since a diminishing of the applied perturbation is assumed at the beginning. At the end we arrive;

$$\frac{\partial}{\partial \Delta} W(\Delta) < 0 \text{ for stability.} \quad (\text{Equation 55})$$

Clearly the bump is stable if the limits of the bump (i.e. the crossing points of the resting level h) are inside the inhibitory range of the kernels (Venkov, 2008). We clearly see that an inhibition type layer in the kernel is strictly required so that the bump does not rise to overall spatial axis or simply disappear.

For a detail version of the above summary and further details on wavefronts, travelling waves or such neural field phenomena please refer to Venkov (2008), Coombes (2005) and finally to Meijer and Coombes (2013).

For understanding the steady state behaviour of human cortical activity, above summarized steady state approach provides a very efficient tool. The only gap of this concept is that the unsteady situations are left out of scope. Such unsteady behaviour are the cases where the system travels from one steady state to another. So passages between behaviour or states are not understood clearly in such approach. But this gap does not create any loss of fidelity in the steady state simulations so it can be used properly to make models of steady state human neural filed activity.

Notice the above given analysis covers the time independent (steady state, equilibrium) solution in the absence of the external input. Now let's look at the subject from a different perspective given by Erlhagen and Bicho (2006). Please remember the state given in

previous paragraphs for the condition where there is no interaction and the activity is only due to the internal dynamics of the field.

$$\tau \frac{du}{dt} = -u(x, t) + S(x, t) + h \quad (\text{Equation 56})$$

Notice such approach makes the connectionist approach impossible since the interaction is left out of scope. Amari approach in fact allows us to construct a connectionist model where the fields are used in a network and the weights of the connections in the network are described with the neural field parameters such as kernel properties. So one shall focus on solving the integral part of the Amari equation to be able to modelling an architecture.

The Heaviside function given previously has a discontinuity on triggering up point so it is not providing an ease of solution mathematically. So it can be preferred using a sigmoid in the form given in previous chapters. Erlhagen and Bicho (2006) suggest a linearized ramp function may also be useful.

Summarizing the findings until now:

$$\begin{aligned} \text{For } u(x) \leq 0 \text{ (i. e. no activity case) and } S(x, t) = 0; \\ u(x) = h \text{ (for } h < 0) \end{aligned} \quad (\text{Equation 57})$$

$$\begin{aligned} \text{For } u(x) > 0 \text{ (for an interval } [0 - a]) \text{ and } S(x, t) = 0; \\ U(x) = h + \int_0^a dx' w(x - x') \end{aligned} \quad (\text{Equation 58})$$

Above approach assumes circularly symmetric and homogene spatial structure for the observed parameter. Until now we have seen the cases where there is no activity and a single bump activity in case of no external input. Imagine the case where there are multiple bumps of activity each having a width of R. In such case;

$$\begin{aligned} U(x) = h + \int_0^R dx' w(x - x') + \int_{x_1}^R dx' w(x - x') \\ + \int_{x_2}^R dx' w(x - x') + \dots + \int_N^R dx' w(x - x') \end{aligned} \quad (\text{Equation 59})$$

The above equation determines the existence, widths and locations of the multiple bumps (Erlhagen and Bicho, 2006).

The field output will depend on a balance between the kernel integral term (i.e. the behaviour of the network) and the strength of the input when an external input is applied to the system (the term S(x, t)). If the input is too dominant (i.e. high) the integral term will lose its effect and the system will output simply the input. Such case is not desired in obtaining a dynamical behaviour. The goal is to have a determining integral term and use the input term only to excite the system. Erlhagen and Bicho (2006) reports, for a robotic application, single pulse and 0 activity solutions are very important and useful. So the inhibitory layers shall be adjusted strong enough to protect the field output from over-strength of the input. Clearly we can see the multiple bump and travelling wave solutions are far more complicated than the single bump case.

3.3.3 Temporal Characteristics of Cognitive Models

The temporal characteristics of the neural models, task timing and sequencing, rhythmicity of human mind and similar timing dependent issues are important and novel study areas of cognitive science and neural modelling. Parallel to the modelling approach discussion between the schools of symbolism and connectionism, temporal characteristics is also subject of another continuing debate.

Clearly depending on their approach to the cognitive modelling, both schools also approach the cognition of task timing with different models. The schools approach to the temporal capabilities of the human mind forms two different approaches to the time estimation modelling. Briefly Taatgen, Anderson and van Rijn (2007) categorize the existing time estimation models under two groups. The first one is named as the internal clock theory (ICT). ICT which is a relatively older theory, claims that there is an internal clock into the cognitive modules of human mind, independent of the task and attention, and similar to other perceptual processes. The internal clock can be modelled with various methods, but the simpler one is to use a clock circuitry which generates continuous periodical pulses, and an accumulator circuitry to count the pulses to measure the timing of any task. Such process may provide the total number of the pulses counted between the end and the start of a task. Such counter can be used to estimate duration of unknown time intervals (guessing). Pulse generator-accumulator couple is one of the simplest approaches to the ICT modelling; there can be other models depending on the process decaying and coincidence detection (Taatgen, Anderson and van Rijn, 2007).

ICT which depends on the relatively older experimental psychology experiments can be modelled with modern symbolic cognitive architectures such as ACT-R. On such architectures depending on the representation manipulation, ICT can be implemented using the capability of the architecture. Notice as mentioned on previous sections, ACT-R includes different buffers such as memory which can be used on implementation of an ICT model (Taatgen, Anderson and van Rijn, 2007). The mentioned reference uses an additional temporal buffer and a temporal module to the original ACT-R to make an ICT architecture.

The second approach to time estimation or counting is the attentional gate theory (AGT). According to the AGT, the task timing is depending on multiple counters (similar to ICT) but the pulse generator is not “strictly” periodic and the pulse repetition frequency of the generator changes with the human focus (i.e. attention) on the task (Taatgen, Anderson and van Rijn, 2007). The main difference between the AGT and ICT is that AGT constructs a relativistic timing concept proportional to the importance assigned to a specific task by the human agent. Taatgen and van Rijn (2011) report another ACT-R based pulse generator-accumulator type ICT model comparison with older experimental data giving acceptable qualitative and quantitative fits of time estimation for tasks shorter than 1 second. Berlin, Rolls and Kischka; giving experimental data on patients with orbitofrontal cortex lesions tending to estimate time duration, claim that the internal clock of such patients may run faster than a healthy human (2004). But notice mentioned OFC lesion experiment is psychological experiment measuring responses of patients and control groups, but does not provide any evidence on the existence of any cortex or nuclei responsible of timing.

Notice the above mentioned ICT and AGT models are more suitable to symbolic models of cognition and for a classical dynamists approach which denies the existence of symbols, the implementation of such “counter” can be problematic. In fact especially ICT

can be found quite opposing the idea of dynamical systems. Dynamical approaches to cognition of time require the understanding of the time relativistically (parallel to AGT) but deny the existence of a clock of pulse generator-counter. For such models, the time shall be estimated relativistically in comparison with task duration but without the counting the time, so a memory of events in spite of a memory of task duration is needed to exist.

A further novel approach, depending on the population dynamics, claims that the time is not marked with individual intervals (i.e. not counted with a clock) but done continuously (Karmarkar, 2011). Therefore timing may not be the function of a specific cortex in human mind, but can be a naturalistic behaviour of the embodied cognition regarding the above mentioned field activity. Karmarkar (2011) names such population dynamics approach as state-dependent-networks (SDN) and describes the state dependent network with following phrases directly quoted from Karmarkar (2011).

Thus instead of marking each interval separately, timing is done continuously, with the network linking multiple signals together as a temporal object. The SDN can only measure information independently, or reset, when the interval between stimuli has been sufficiently long to allow the network to return to its baseline state (Karmarkar, 2011, p.1).

Notice above claim is very compliant with the DFT activity and working memory approaches. Karmarkar's SDN returns to its baseline state similar to a dynamic field returning to its inactive state in the lack of the external perturbation. In another study, Spencer, Karmarkar and Ivry (2009) declare time as an emergent property of dynamics of the SDN and reject the clock-counter models. These studies although naming the networks as SDN, still provides a supporting basis to the temporal approach of the DFT based modelling. Karmarkar and Buonomano (2007) supplies experimental data on the existence of two different mental utilities of temporal measurement, one for the millisecond processes and one for second processes. i.e. Their claim is estimation or cognition of the timing for tasks with shorter duration (<1 seconds) seems to be compliant with SDN thesis, and longer tasks modelling (>1 seconds) seems not benefiting from SDN approach. Also there might be a intersection of both sets (600-800 msecs) working with both models.

Any way regarding above supporting ideas, DFT's approach to temporal activities may be found acceptable. DFT assumes that the timing of each task and task sequencing is done by the network activity patterns, i.e. no measurement is done, and there are not any discrete counters. Such approach is clearly parallel with above listed findings opposing ICT and AGT. DFT provides a framework where everything should be done in continuous and embodied ways. This is completely compliant with dynamic modelling approach which does not use any algorithmic approach and does not contain functional metric timers. This is the primary approach of DFT. Another claim by Leon and Shadlen (2003) depending on experiments with macaques; is that there may be a specific part of cortex dedicated to comparing task durations with each other to decide one is shorter than the other or vice versa. Still this measurement is not metric but it is dynamic. Therefore DFT may provide a field comparison capability compliant with Leon and Shadlen (2003) claim since the claim does not require any discrete counter requirement. It shall be noted that the researches on the temporal functionalities of human mind is at a very early phase and none of above studies may provide a complete understanding of our time awareness or task time management functions. Even the DFT may seem providing a parallelism with

some modern studies, it is also contradicting with various some other modern studies mentioned below.

Looking from a modelling side, representational or computational architectures such as ACT-R uses the architectures declarative memory to hold recorded temporal values to compare for later events (Taatgen, Van Rijn and Anderson, 2007). Using such method ACT-R achieves very successful results in some multitasking experiments (Taatgen, Van Rijn and Anderson, 2007; Taatgen and Van Rijn, 2011). Notice DFT approach does not need such a task. Each dynamics shape its own temporal behaviour. If such a task sequence shall be arranged regarding temporal characteristics, a sequential organization capability is needed and it is achieved in DFT via ordinal dynamics logic which will be briefly summarized in later sections.

Notice on above paragraphs only a comparison between the representational architectures approach to temporal capabilities with DFT is given. But in fact neurally based modelling is not limited with DFT as it is known from previous paragraphs. There are LIF/INF based network models of temporal capabilities. Regarding the requirement of the complexity such models can be constructed. One good and successful example can be found in Medina, Garcia, Nores, Taylor and Mauk (2000) which use a unified model of LIF-like neurons network and synapses to analyse temporal capabilities.

On this thesis, an architecture based on DFT will be used to perform piloting tasks. In case of need for any temporal capability task dynamics will be preferred compliant with existing literature in spite of discrete time counters. Especially the “ordinal dynamics” which is used to adjust task hierarchy and sequencing is one important temporal capability supplied by DFT literature.

3.3.4 Task Hierarchy and Sequencing in DFT

DNF/DFT studies and robotics application in many cases require performing of more than single task to achieve desired target behaviour. For example navigation from one point to another one in the space requires sensory tasks, motor tasks, decisions (such as direction and distance) to be performed hierarchically, serially or parallelly. Another example may be getting a cup of water from the table. To acquire such goal, one shall locate the glass, adjust the arm angles and distances, reach the glass and grab it. Clearly most of the tasks on the daily life, in fact are sum of multiple parallel or serial sub-tasks. So to achieve the goal of cognitive modelling of a complex task such as piloting, a method of task sequencing and ordering will surely be needed.

Most of the dynamic neural field based cognitive modelling studies uses an approach named as ordinal dynamics (OD). The method can be summarized with following properties.

OD is a method where tasks or sub-tasks of a major task are organized ordinally in a serial way. Each task which is in the ordinal stack is named as an elementary behaviour (EB). EB's are formed of intentions and decisions. Each EB has its own intentionality layer symbolized with a dynamic field named as intention field. Additionally after performing the task, a layer is used to understand if the EB intention is reached. This field which is used to decide if the intention is needed and acceptability of switching to the next task; is named as constraint of satisfaction field (CoS). Therefore each EB is formed of an intention and a CoS field. When the CoS is activated, it means the next task is to be performed. So the ordinal dynamics decision to switch to next task is activated via the input from the previous task's CoS output (Sandamirskaya, Richter and Schöner, 2011;

Kazerounian et al., 2013; Sandamirskaya, 2011; Sandamirskaya and Schöner, 2010, 2010b, 2010c). In case of the intention of the task is to perform an action, an action field can be added to the architecture for that specific task. i.e. the architecture complexity may be increased when required.

The OD is a continuous and dynamic field model (as all the DFT based model) but may produce discrete outputs to activate the next task operation. So the equation set for OD shall be in the form of Amari equation since it is a dynamic neural field but shall also be compliant with the goal of producing predefined discrete outputs. For achieving such property, the OD model is structured depending on “nodes” of discrete activity which provides perturbing signals to the fields. So discreteness and continuity of the OD is shared between the nodes and the fields of the corresponding EB. On the following equation the dynamics of an OD is given (Duran, Sandamirskaya and Schöner, 2011).

$$\begin{aligned} \tau^o \frac{dv_i^o}{dt} = & -v_i^o(t) + h^o + c_i^{o,o} f(v_i^o(t)) - c_i^{o,s} f(v_i^s(t)) \\ & + \mathbf{W}_i^{o,m} f(v_i^m(t)) + \mathbf{W}_{i-1}^{o,o} f(v_{i-1}^o(t)) \\ & - c^- \sum f(v_i^o(t)) \end{aligned} \quad (\text{Equation 60})$$

Although above equation 60 might seem different, in fact it is clearly a discretized version of the Amari equation. On below list each item of the equation 301 is described clearly (Duran, Sandamirskaya and Schöner, 2011).

τ^o : Time constant of the field dynamics. This parameter describes how fast or slows the system state changes.

h^o : Negative resting level. This parameter contributes to the stability region of the field activity.

$c_i^{o,o}$: The strength (amplitude) of the self excitory input.

c^- : The strength of the mutual inhibition between the ordinal nodes.

$c_i^{o,s}$: The strength of the inhibitory input from CoS node. This parameter increases the amplitude when the corresponding CoS node is active and so the ordinal node becomes subject to a inhibition and node state decays to inactive.

$\mathbf{W}_i^{o,m}$: A matrix of weights for connections with the memory nodes. This matrix controls the sequential organisation of the ordinal nodes.

$\mathbf{W}_{i-1}^{o,o}$: A matrix of weights for connections with the ordinal nodes of upper layer. This matrix is used in hierarchically organised nodes. For sequential dynamics such matrix is not needed.

$f(v_i^o(t))$: A threshold function. Notice this time the threshold function is not convolved with the kernel, but it is multiplied directly with the input strength.

When the subject is not a node but a field, previously described Amari equation with convolution integral will still be valid. The only difference will be the field dynamics acquisition with multiple convolution due to the addition of the connections between the CoS and intention fields. On such architecture more than a single kernel may be used depending on the structure. When needed or the architecture allows, the node equations

can be simplified much in a form which allows very easy numerical solutions (Richter, Sandamirskaya and Schöner, 2012).

$$\tau^o \frac{dv_i^o}{dt} = -v_i^o(t) + h^o + c_i^{o,o} f(v_i^o(t)) \quad (\text{Equation 61})$$

Notice the above equation 61 includes only the self excitory strength parameter, so many of the nodes connections are eliminated. Although the node dynamics is summarized above, the CoS or intention field dynamics will not be given here since it is simply the Amari equation. The only difference is an additional term symbolising the connections with the ordinal nodes $\sum f(v_i^o(t)) \mathbf{W}_i^{intent}$.

In the following section, some examples from the robotic applications developed using the DFT based architectures will be briefly explained. Most of the robotic application which will be described on the following paragraphs contains OD based architectures for sequential task or task hierarchy. Although the method has very successful robotic implementations in the literature, there are critics of some aspects of the method. The OD method depends on a sequential memory of tasks, i.e. all actions to be performed and the intentions belonging to such tasks shall be already hold (recorded, predefined) in the intentional layers of each elementary behaviour and task order shall already be defined and known. In such approach, ordinal dynamics method which is a popular task sequencing tool of DFT needs the task order is learned previously. Muratov, Lakhman and Burtsev (2014) criticise the DFT, claiming that there is a lack of capability on detection of the dynamical environment or autonomous exploration. The mentioned reference contains usage of a network of simple neurons to form a *neuroevolution* in the exploration of unknown environment. Although the neuroevolution method may be successful for solving relatively low number of neurons, in case of huge networks are subject, DFT approach will have a very big computational resource requirement advantage. Notice for relatively smaller networks, DFT may be disadvantageous due to the complexity of the neural field equation solutions.

3.3.5 Robotics Applications of DFT

On previous paragraphs, analytical solutions to Amari equation for single bump cases steady states are given. Notice such solution needs a discretization for solving in the processor of robotic application.

For robotic spatial task intelligence or decision architectures, the activation of the neural fields is simply the representations of spatiotemporal decisions or solutions. In such case the solution/activity $U(x, t)$ depends both time and the spatial variable. For such situation Erlhagen and Bicho (2006) provides a discretization scheme constructed upon equally sampled discrete points on x axis. Then the solution to the Amari equation can be performed via a simple integration operation. Multiple numerical methods can be used but Erlhagen and Bicho (2006) prefers usage of forward Euler method. Below is the discretization scheme for Amari equation and integral method by the mentioned reference:

$$x_j \doteq x(j \Delta x_d) \quad (\text{Equation 62})$$

$$u_{j,i} \doteq u(j \Delta x_d, i dt_{Euler}) \quad (\text{Equation 63})$$

$$S_{j,i} \doteq S(j \Delta x_d, i dt_{Euler}) \quad (\text{Equation 64})$$

$$w_j \doteq w(j \Delta x_d) \quad (\text{Equation 65})$$

$$u_{j,i+1} = u_{j,i} + \frac{dt_{Euler}}{\tau} [-u_{j,i} + S_{j,i} + h + \sum_{k=0}^{N_j} w_{j-k} f(u_{k,i}) \Delta x_d] \quad (\text{Equation 66})$$

$j=0, 1, 2, 3, \dots, N_j$

The symbol \doteq is used for vector equality. The equations 62, 63, 64 and 65 just show the indexing of the components of discrete nodes forming the space for spatial axis, activity, input and kernel. On above equation set Δx_d is the sampling interval i.e. spatial resolution on the x axis. So the system will model the spatial axis x with steps of Δx_d . The index i which is used o above set represents the discrete time events. dt_{Euler} term is the time step for Euler integration. Equation 66 gives a discrete solution to Amari equation, where the activity for a specific time $i+1$ is computed using the activity $u_{j,i}$, input $S_{j,i}$ and resting level h of previous time step i and the kernel w_{j-k} over space. Function $f(u_{k,i})$ is the threshold function. The Euler integration time step shall be much smaller than the time constant τ for that the dynamical behaviour of the Amari equation can be observed and the given discretization works ($dt_{Euler} \ll \tau$) (Erlhagen and Bicho, 2006).

As previously mentioned, the biggest advantage earned by the dynamic approach of cognitive modelling is that the system is embodied. The embodiment and the situatedness of the DFT approach ease the development of dynamic cognitive robotics since the sensor-agent-world coupling is accompanied by the nature of the agent. On a time period bigger than a decade, the literature of robotic area presented various studies using DFT techniques to embed spatiotemporal task oriented intelligence to cognitive robotics. Additionally bio-inspired robotics applications can be simplified using the DFT aids. Erlhagen and Bicho (2006) provides a very detail clarification on the advantages of the neural field usage in robotic intelligence development with providing similarities between human cortical functions (such as working memory, superior temporal sulcus or prefrontal cortex) and DFT models. For further information on such relations please refer to that study. Notice the embodiment is an important requirement of robotic development targeting dynamic environment. For further information on present robotic architectures including DFT based ones, interested reader may refer to Hemion (2013).

Bicho, Mallet and Schöner (2000) provide a cognitive robotics application using DFT for motion direction decision layer which uses multiple low level sensors such as InfraRed sensors and microphones. On referenced study, simple sensors are used to shape a situational awareness of targets, environment and obstacles. On such environment with complex targets and obstacles changing position, it is shown that DFT can provide cognition and, as stated on the last paragraph of previous section, due to the simplicity of the approach the probability of forming a bottleneck is quite low in comparison with other methods of artificial intelligence. The mentioned study uses the attractor dynamics of the sensory fields on generation of the behaviour. Memory functions are also developed using neural fields (Amari equation on that example) with relatively longer decaying time.

Cognitive robotics studies do not focus only to single robots but also to robot-robot and robot-human agents' interaction. Monteiro and Bicho (2009) report a study of 2D robot navigation for multiple agents performing formation missions. The study contains both

the computer simulations and implementation of the claimed algorithm into Khepera commercially available robots. The main point here is that neural field approach to such multi-agent navigation planning provides flexibility such that Monteiro and Bicho could provide three different methods on the development of the formation between robots. Attractor dynamics under the control of the neural fields just provide an infrastructure for robots cognition in flexible manner so that different methods can be developed using it.

Similarly Bicho, Louro and Erlhagen (2010) provided robot-human interaction studies where implementation of neural field intelligence used in cognitive robots. One strong side of the mentioned study is that it provides verbal and visual interaction of the robot with the human agent sharing the same environment. The study reports the results of a joint construction task performed by a robot and a human. Assuming that the future robots will share their living space with human and will serve the humans, the interaction of natural and artificial agents is one of the most promising study areas.

Notice when we speak of robotics, the general understanding is that there is a body on a physical environment. Alternatively the bots (with other words software bots, soft bots, and software robots) which are simply working on soft environment may at first look seem to be out of the field. But in fact a bot is also interacting with the simulated environment in soft world and it can simply assume being a tool of development for physical robotics with various other benefits. On that thesis given here, the subject will be a pilot intelligence in form of a bot interacting with a model of aircraft and environment.

Schöner, Dose and Engels (1995) summarize how to implement DFT based architectures on robotic applications via describing the link between a system's dynamic equations and its behaviour i.e. the study covers the relation between the attractors and the environment. The mentioned study uses a commercially available and programmable robot platform MARVIN and implementation of two different navigation orientation tasks. On literature there are various examples of commercially available robots (e.g. above mentioned MARVIN and Khepera, and ARoS used in Bicho, Louro and Erlhagen (2010)) usage with DFT examples. One reason of that various platform with various operating systems have usable with DFT is that there exist very simple numerical solutions for steady states of Amari equation.

Interested reader may refer to Sandamirskaya (2013) for DNF/DFT connection with the functional layers of human cortex on robotic applications where a good introduction to field manipulation in robotic architectures is given. Sandamirskaya in that study reports a VLSI (Very Large Scale Integrated Circuit) embedded implementation of a neural field based robotic application. Regarding the previous concerns given on above paragraphs about the computational resource problem on an autonomous robot platform, VLSI integration can be an efficient solution. Sandamirskaya shows a similarity between a soft Winner-Takes-All (WTA) network and neural field based architectures. So the mentioned study is focused on WTA implementation aspects on robotic towards the goal of developing a neuromorphic architecture by DFT.

3.4 Frameworks

3.4.1 Cosivina

Cosivina is a MATLAB® based framework to model DFT models of cortex and displaying outputs developed mostly by Ruhr University in Bochum, Germany. At early stages of this study a Cosivina version supplied by Ms. Yulia Sandamirskaya from above

university is used to observe different behaviour of neural fields with varying parameters. MATLAB version 2010a is used on implementation, running and evaluation of Cosivina based models.

3.4.2 CEDAR

CEDAR is a Linux based framework providing infrastructure to fastly develop embodied cognitive DFT models. In this study CEDAR version Crimson King is used as the primary modelling tool (Lomp et al. 2009). CEDAR application is run on Linux Ubuntu 2.1. The framework provides the user with a graphical interface where many DFT tools are available and can be used with drag-drop methodology to derive cognitive models. The framework allows the user to build up externally compiled software extensions as plugins added to CEDAR. Virtual interface provided by CEDAR has been very useful during the study. For further information on CEDAR and capabilities please refer to Lomp et al. (2009).

Jsonlab.1.0beta is used on recording and observing the model parameters since it is easily interfacing with CEDAR and the interface is already implemented inside CEDAR.

On below figure a CEDAR screenshot is provided. The biggest advantages of CEDAR are that it provides drag-and-drop filed modelling and connection, and instant graphical observation of multiple fields at the same time. The figure contains a summary of the model architecture while including the sensor or perception block, the decision or with a formal name the executive block, the motor block and finally the aircraft model and the 3 channel connections in between these blocks.

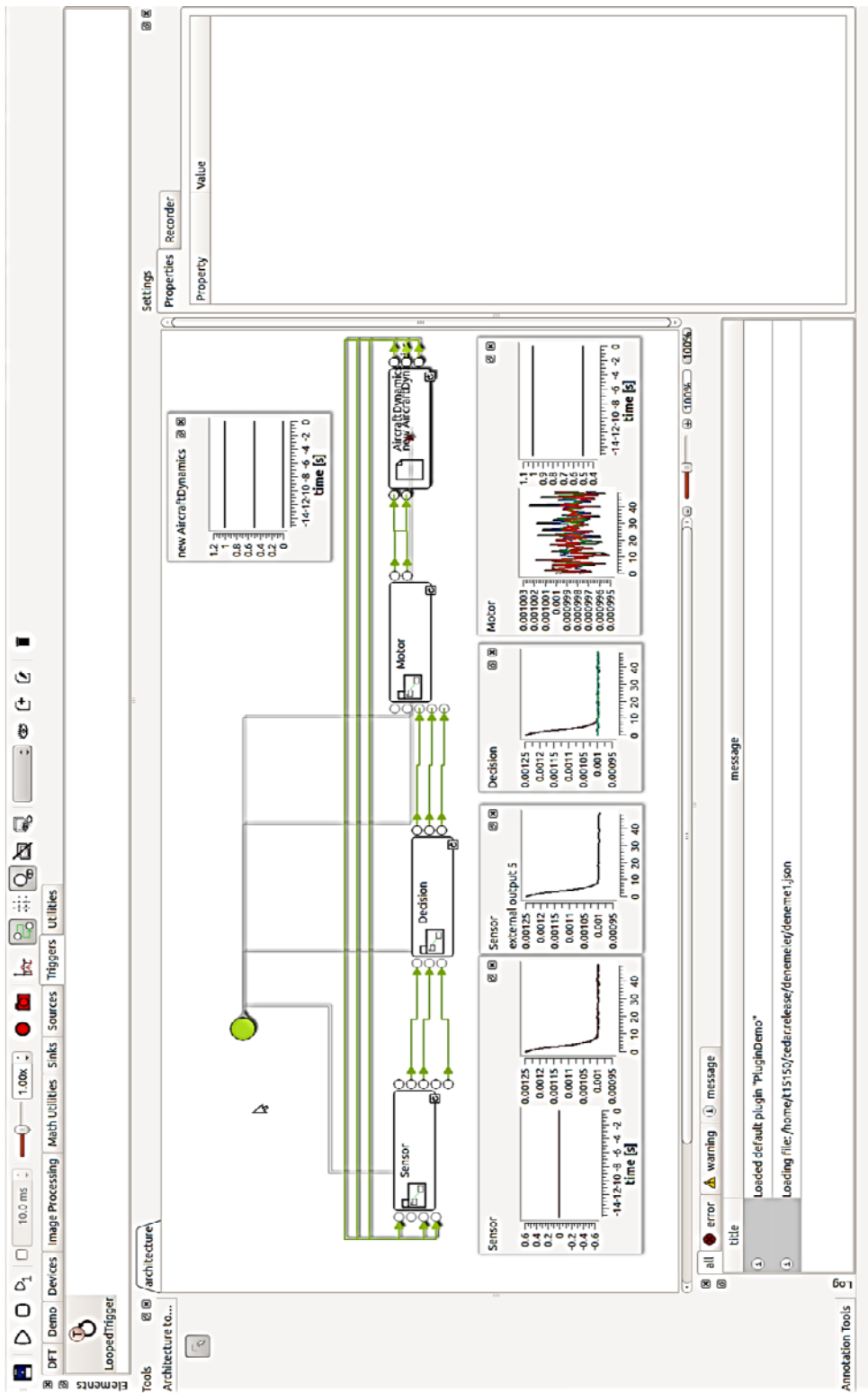


Figure 24. CEDAR Screenshot.

This page is intentionally left blank.

CHAPTER 4

4 STRUCTURE OF THE MODEL

On the development of the final model, a 3 steps approach is used. On the first model the aim was only to perform a continuous level hold performance for an aircraft on 2 axes. The roll axis control is neglected for simplicity. On the second step, the first model is modified by adding a modification to the intention layer. On below paragraphs both models will be summarized. Finally on the 3rd model an improvement is made to provide a model with varying performance under changing altitude.

The models subject to thesis are formed of two integrated parts. First part is of course the DFT based pilot model. The second part is an aircraft model which will be controlled by the pilot model. These two models will interact each other in the modelled environment.

4.1 The Aircraft Model

In literature there are various types of aircraft models. For advanced design studies or analysis where the detail characteristics of the aircraft are important, generally a 6 degree of freedom (6DoF) model is used. Regarding the purpose of the analysis and modelling, the degree of freedom can be decreased to 5 or 3. Such models are also commonly used. In this thesis, main goal is to construct a pilot model. Regarding this fact, an aircraft model with 3DoF is written and used. For later trails a 6 DoF model may also be used. In below paragraphs details of 3 DoF aircraft model is given.

4.1.1 3 DoF Aircraft Model

3DoF models are generally used in analysis where the phenomenal and dynamic behaviour of the aircraft are not required to be understood in detail. In other words; when a researcher is focused on understanding the phugoids, inertia tensors, oscillations or moments to control an aircraft, the constructed model shall include as much as possible degrees of freedom. So in such case a 6DoF model with realistic inputs are required. But when phenomenal or short term dynamic behaviour of the aircraft is not needed, lower degree of freedom models may be useful. The selected 3DoF model in that thesis fits very well for understanding long term navigation behaviour and similar models are used in some previous studies (Carretero, Nieto and Cordon, 2013).

A 3DoF model neglects the moments inputs of the aircraft model, does not contain angular rates modelling and Euler rates. Our model will include a point mass aircraft with

a linear drag model. The mass of the aircraft is decreasing proportional to the thrust used. Below is the mathematical summary of the 3DoF aircraft model. On the below figure a representation of some angles and axis is shown for better clarification of the details of the model.

$$\frac{d}{dt} \begin{bmatrix} X \\ Y \\ h \\ V \\ \varphi \\ \gamma \\ W \\ T \end{bmatrix} = \begin{bmatrix} V \cos(\varphi) \cos(\gamma) + w_x \\ V \sin(\varphi) \cos(\gamma) + w_y \\ V \sin(\gamma) + w_z \\ \frac{g}{W} [(T \cos(\theta - \gamma) - D) - W \sin(\gamma)] \\ \frac{g \sin(\rho)}{W V} [L + T \sin(\alpha)] \\ \frac{g}{W V} [(L + T \sin(\theta - \gamma)) \cos(\rho) - W \cos(\gamma)] \\ -C T \\ 0 \end{bmatrix} \quad (\text{Equation 67})$$

The below matrix given in set of equation 67 contains the rate of change by time of the fundamental model parameters. Below is a clarification of each parameter used on the matrix.

X: Longitudinal axis position of the aircraft, i.e. along body axis position. The term dX/dt defines the rate of change of *X* position by time in longitudinal. Notice this position data is in a flat earth Cartesian coordinate system where an initial position can be given to the aircraft. For the simplicity, one easy method can be to put the aircraft in the origin of the coordinate system.

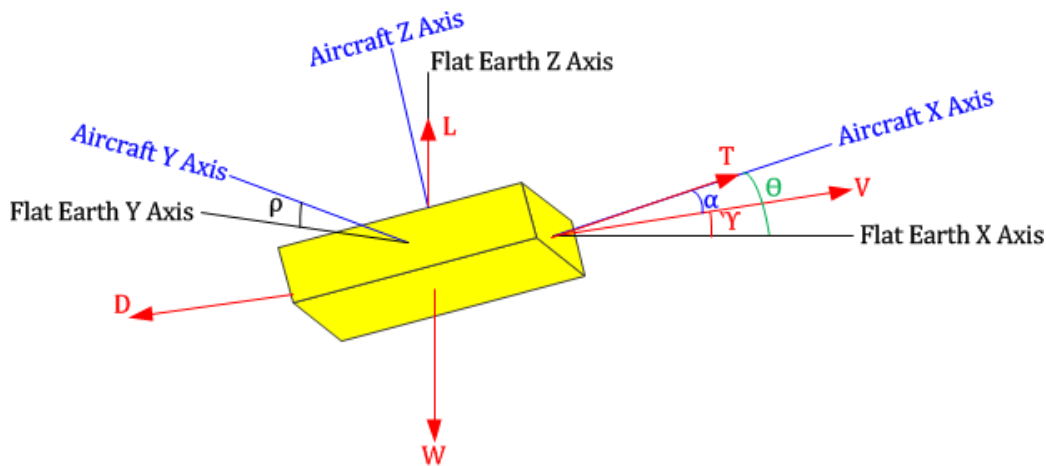


Figure 25. A Representation of a flying body neglecting the rotational forces for 3DoF motion model. Axis and angles are summarized for 3DoF body. The heading assumed being North, so Φ is not shown.

Y: Lateral axis position of the aircraft, i.e. cross body axis position. The term dY/dt defines the rate of change of *Y* position by time in lateral. Notice this position data is in a flat earth Cartesian coordinate system.

h : Altitude of the aircraft upon flat earth Cartesian coordinate system. i.e. the normal axis position. The term dh/dt defines the rate of change of altitude by time.

V : The speed of the aircraft upon flat earth Cartesian coordinate system. Normally the speed has components in 3 axes. Such terms like $V \cos(\varphi) \cos(\gamma)$ gives the components of speed in different axis. dV/dt is the rate of change of the speed in time. Notice rate of change of speed is the acceleration and is a function of drag, weight, aircraft angles with earth surface and gravitational acceleration.

φ : Aircraft current heading angle. The term $d\varphi/dt$ defines the rate of change of the heading angle. The heading angle is measured from a North up reference system. i.e. 0 of heading is assumed as the north direction.

γ : Aircraft flight path angle. This angle is the angle between the flat earth surface and the aircraft velocity vector. The rate of change of that parameter by time is shown with $d\gamma/dt$.

W : The weight of the aircraft plus the weight of the fuel. In normal conditions, fuel is expected to decrease proportional to the thrust applied by the engines to the aircraft. In this example, thrust is held constant, so the decrease in the W is linear with a constant term C named as the specific fuel consumption coefficient.

T : Constant aircraft thrust supplied by the engines.

g : The gravitational acceleration. Normally g value changes regarding where you are on the earth surface. But when a flat earth model is used, g can be assumed constant.

w_x, w_y, w_z : Components of wind vector. i.e. the speed of wind in North and East direction and finally the third one is in normal axis.

θ : Aircraft pitch angle; the angle between the flat earth surface and the aircraft nose. Notice this angle is different than γ , since this is not considering the velocity vector but takes the angle between the main chord line and the flat earth surface.

ρ : Aircraft roll angle; The angle between the lateral axis of the aircraft and the flat earth surface.

L : The lift force. In this model it will be taken equal to a proportion of W with a coefficient of $(1/\cos(\rho))$. Please see equation 202.

D : The Drag induced on the body of the aircraft due to the airspeed. It can be taken as a direct proportion of lift or can be a linear or nonlinear function of speed. In this model a linear relation between the drag and the lift will be constructed regarding the V of the aircraft. Please see equation 69.

α : Angle of attack; the angle between the aircraft mean chord line (i.e. longitudinal axis of the aircraft) and the velocity vector.

Sideslip or skid effects are all neglected. Lift is considered equal to vertical component of the weight for level flight. Thrust is taken equal to the weight in the very first trial. For each trail, applied model parameters will be listed in the relevant paragraph. Finally a 0.05 Lift to Drag ratio is assumed for 200 Knots of airspeed so that a coefficient equal to $10^{3/4}$ is used. While the speed of the aircraft changes the L/D ratio will also change. Please see below equations 68 and 69.

$$L = \frac{W}{\cos(\rho)} \quad (\text{Equation 68})$$

$$D = \frac{V * 10^3}{4} * L = \frac{V * 10^3}{4} * \frac{W}{\cos(\rho)} \quad (\text{Equation 69})$$

Above given model is implemented in C++ and connected to pilot model which is implemented in CEDAR framework using a DFT approach. A python code for the 3DoF aircraft model will be given in the appendices of this thesis to provide a simple implementation of the model to show how the model works to the interested reader.

4.2 Step 1: Constant Intention Pilot Model

To be able to interact the pilot model with the aircraft model, the aircraft mathematical models written in C++ are built up as CEDAR plugins inside the framework. To create an interface between the pilot and the aircraft, the aircraft models inputs and outputs shall be acquirable by the pilot. In a real aircraft such functions of interfacing, i.e. the perception of the aircraft situation, are performed via embodied sensors of human agent. Such sensors can be the eyes with the support of the visual nerves and visual system tools such as Lateral Geniculate Nucleus (LGN) and Striate Cortex including V1 to V4 cortices. Notice a biological sensor can be very complex. Here the aim is not to model the sensor but to provide a model of pilot cognition with similar performance. Even CEDAR framework includes tools to use as sensor interfaces; here the construction of a functional interface is preferred.

The aircraft model provides amount of deviation from level flight. This classically what a flight display does. A normal attitude and direction indicator or namely a primary flight display includes visual cues to demonstrate the aircraft attitude deviation from level. Similar observation can be done by the inertial sensor of the human. With our personal experience, we know that experienced pilots are able to feel the aircraft attitude changes many times. So whatever the sensor is, the pilot acquires the deviations from level flight. So the supplied variation numerical values may be transformed into space code for DFT model. Since CEDAR neural fields accepts CV-32F data format aircraft models are written as providing the 3 axis deviations in this format. The ΔX , ΔY and Δh values are transferred in to space code using the Rate to Space Code transformation allowed by CEDAR.

The space codes are easily acquirable by neural fields since those primary representations of spatiotemporal environment. For all 3 axis deviations, the space codes are connected to single dimensional neural fields to construct the perception threshold of the values. The perception neural fields provide thresholds for each axis so that, temporal behaviour of the acquisition can be generated by a delay. Additionally, the fields' behaviour under very slow perturbations such as noise can be used to construct a filter against noise. Notice in normal situational awareness conditions, the pilot does not produce control movements to diminish the aircraft vibration. So here shall be a threshold which had to be crossed to start the control process. Such threshold is built up by using the negative resting level of the fields and the inhibitory parameters. Please refer to below drawing to see the perception architecture.

Leftmost block of columns in below figure represents a cognitive function regardless of its biological host cortex, performing the understanding the deviation of the aircraft attitude from level position. Notice such attitude deviation acquisition or perception

would also be implemented by a 3D neural field. But a 3 line implementation in components of one dimension simplifies the analysis. Therefore a component level approach is preferred in spite of using a 3D matrix. Notice component or matrix wise, whatever the approach is, deriving the spatiotemporal meaning of any sensory cue is a very complex task and is generally performed by strong parts of human brain such as V4 or VTC visual cortices in Striate cortex.

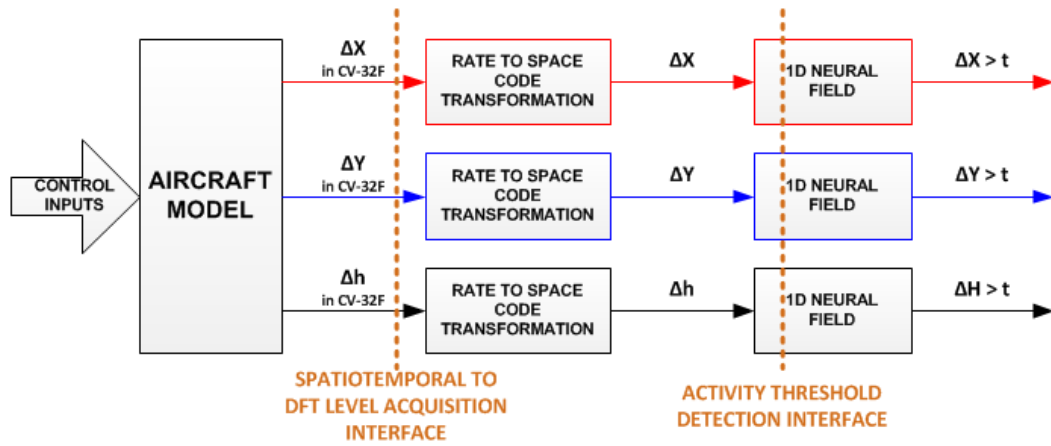


Figure 26. Acquisition/Perception Architecture. This part of the pilot model is responsible from interfacing with the aircraft model, transforming the acquired rate data (let us say a scalar introduced in form of a CV-32F matrix) into the space code, perception of the space code of each dimension with one dimensional neural field and using the field as a detection of threshold. The architecture creates an activity only when the field input is above a selected threshold. Such threshold can be used to implement a temporal behaviour and eliminating perturbations.

Notice, after this point the aircraft attitude is known by the model and shall be processed to produce a control decision. Decisions are generally most important parts of neural modelling tasks because the behaviour of the model is as expected generated regarding these decisions. Below figure summarizes only one channel of the decision layer namely the lateral correction channel. The other two channels will be similar to below represented.

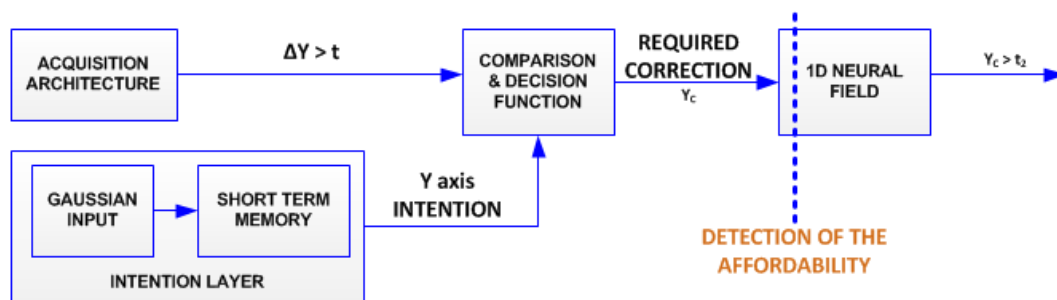


Figure 27. Decision Architecture in Lateral Channel. Only one channel is shown. Other two are neglected since similar to this one.

The decision channel needs to include an intention layer which is used to memorize the goal of the model. In this very first example our model is only aiming to hold the aircraft

n level attitude, so our intention layer requires only memorizing “0” in all axis. The Gaussian input feeding the short term memory is a transient wave in form of a Gaussian with centre at “0”. Once the Gaussian is activated and then decayed or disintegrated, the short term memory block requires remembering this value for an acceptable amount of time which is adjusted by time constants of the field. Originally the short term memory is also a neural field with a very high time constant creating a very slow decaying behaviour. With other words decaying time constant of a neural field can be adjusted to create the speed of forgetting any data. Probably the intention layer is the heart of each cognitive model, because one can construct a dynamic intention layer to shape a dynamic cognitive model. If one can update the intention layer with different hold pattern values, the model can perform different task with the aircraft under its control. The second step of pilot model includes a changing intention layer and it will be described after this one.

Notice similar to the perception fields on previous blocks of acquisition, decision architecture contains various neural fields. One of these fields serves the memory function to hold a space code intention. The second on is the affordability detection block which decides the required amount of correction is achievable or not. When the required amount of correction is below a minimum threshold such as below muscular tolerances or above a limit such a limit of human muscular power, the decision may be not to create any activity. The other neural fields are hidden inside the comparison and decision block; these fields are used to produce a subtraction operation between the intention and the deviation to decide the amount of correction. The subtraction can be easily performed via using the interaction of one excitory and one inhibitory neural field. Finally, the deviation value is in any units of spatial measurement, here in feet. But the correction shall be in units on angular displacement since the attitude values of an aircraft is controlled by angular correction caused by control surfaces created moments. So the decision and comparison block contains some unit conversions, i.e. a mapping between the spatial displacement correction requirements and the angular equivalent of such value. It shall be added that, in each of the arrows showing the direction of the information or activity flow in above figures, lines can include static gain functions to make detail signal level adjustments. Such static gains are similar to neurotransmitters increasing electrical activity in one direction inside neural network.

When the decision is given for the amount of the correction angular action, it shall be performed by muscles, so it is transferred to the motor functions. Motor functions include the acquisition of the decision and the mapping between muscular tension and the decision angle. So it is a relatively easier transformation unless multiple angles are omitted. In this study only one angular tension is used on motor function. Below is block schematic of motor functions.

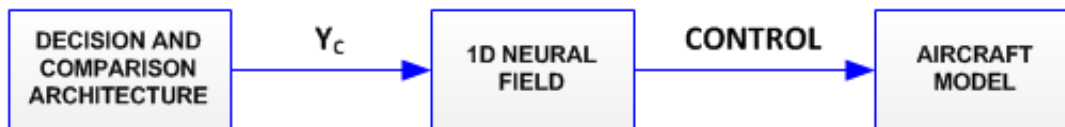


Figure 28. Motor Behaviour Summarized.

Notice to decide an angular correction on axes, above architectures shall be unified. Below is representation of such unification in CEDAR blocks format. All three channels are shown in parallel. As previously mentioned, parallel channels usage is not an obligation but is a preference for simplicity. Normally, to decide a lateral deviation

correction, one shall notice the speed of the aircraft or in other words the amount of forward movement in unit time. Therefore different than the above simple parallel channels, normally lateral control channel requires both the ΔX and ΔY values to provide a realistic solution. Similar idea is valid for altitude deviation correction requiring ΔX and Δh usage in a synchronised way. In fact pilot model shall contain information passages between channels to provide an altitude and lateral situation (i.e. heading) correction solution.

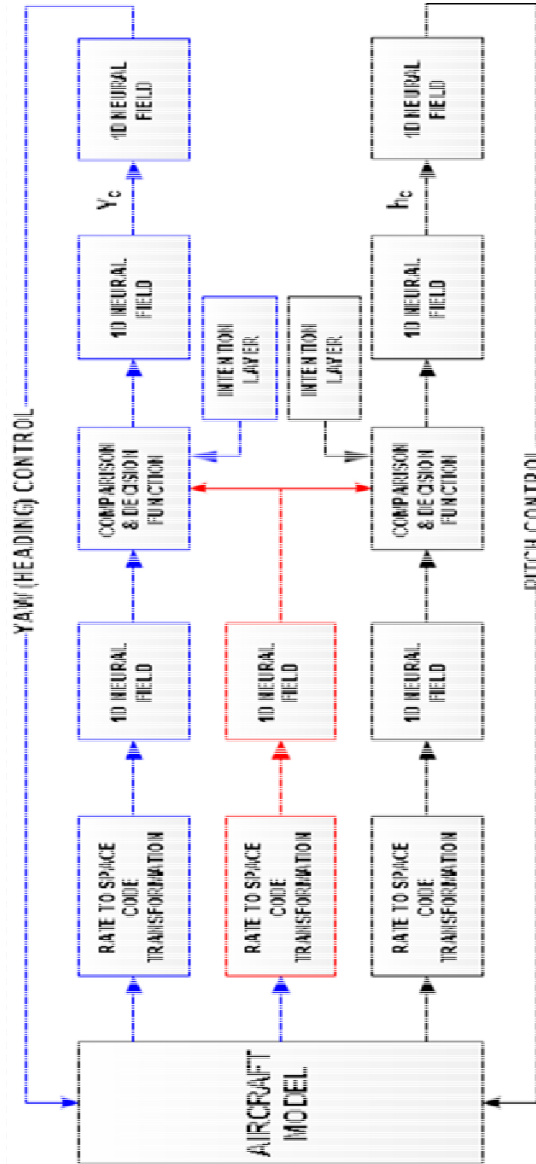


Figure 29. Complete Step 1 Model Architecture. Step 1 pilot model creates required amount of pitch and heading correction inputs to hold the aircraft on level attitude. Only the roll channel omitted for sake of simplicity. Level attitude goal is stored in each channels intention layer which contains a short term memory.

The above architecture is modelled using CEDAR and various simulations are run. For results see relevant chapter below.

4.3 Step 2: Improved Pilot Model

In a regular flight, pilots control pattern is not only formed of hold patterns. If this were the real case, pilot would be replaced simply by an autopilot. But even for UAVs (Unmanned Aerial Vehicles), in fact pilots are in the control loop due to the critical decision requirements. There are autonomous modes but these modes covers performing of some routine functions such as running the aircraft in a racetrack pattern or holding in an altitude. So a cognitive pilot model shall include much more than a cognitive autopilot.

When above step 1 architecture is analysed, one will notice that the critical item to provide a goal to the pilot model is the intention layer. For being able to design a model which will perform varying tasks in an order, predefined or not, the intention layer shall be much more complex than a short term memory. The short term memory in each channel shall be modified regarding the goal. There are various ways of doing such improvement in DFT area and CEDAR includes some useful tools.

The easiest way of doing such a modification is using the ordinal dynamics methods. Ordinal dynamics approach provides a discretized Amari equations set forming nodes of activity to start different tasks. The completion of each task is defined with a “Constraint of Satisfaction” node and the ordinal dynamics (or serial dynamics) gives start to next task. For further details please refer to relevant chapter above. Ordinal dynamics blocks can be used to create hierarchical task dynamics by adjusting relations between various serial dynamics. A simple diagram is given below.

The main idea of serial dynamics is that each task is started after the ending of the previous one in the serial order. This temporal behaviour is very compliant with previously mentioned temporal behaviour measurements and models by Karmarkar (2011). But when one focus on the piloting behaviour, a different task scheduling will be observed. The pilot behaviour includes a continuous check of external dangers such as traffic or ground collisions and it is quite acceptable that in a pilot tasks list, the obstacle avoidance has the highest priority. So in case of an obstacle is observed, the intention layer shall be updated with “numerical intentions” to save the aircraft from collisions.

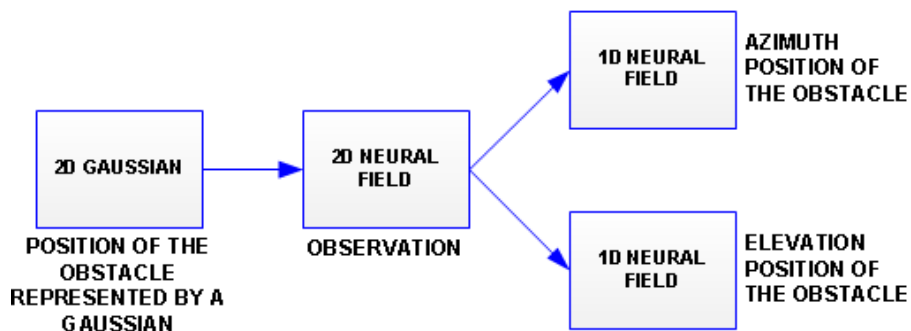


Figure 30. Obstacle Position Perception Architecture. The given blocks are used to determine the azimuth and elevation positions of an obstacle from a 2D screen.

An obstacle shall be first observed to create a change in the intention layer. So an “observation architecture” will be needed. Pilots view from the front window of the canopy or a display screen including external camera view contains a 2D plane where the aircraft is aligned towards the centre of it. Clearly a two dimensional detection would be

required. Then the 2D observation shall be divided into two components of 1D since the rest of the model is planned as components. Please see above figure 30.

When the azimuth and elevation positions of the obstacle are obtained, the data can be used to generate an update in the intention layer. But there shall be “detection switch” to decide if the obstacle exists or not, so that in case of “not” the intention layer easily returns to 0 for keeping the pilot model holding the aircraft at level flight. The detection switch mechanism is kind of logic gate implemented in CEDAR as mathematical function. It is very similar to a weighted exponential filter. Please see below figure and following equation to understand the detection switch mechanism.

On the below figure 31, 0D neural fields are used as activity detectors to notice the existence of an obstacle. Such fields may be used as unitless demonstration of an activity. A summer is used as an OR gate to create a single activity detection signal. Finally a switch function is used to produce the intention. Switch function is given in below equation 70.

$$Int = (1 - k) * i2 + k * i1 \quad (\text{Equation 70})$$

$$D = X_{obs} + X_{ac} + \partial \quad (\text{Equation 71})$$

This switch function makes a choice between the inputs $i1$ and $i2$. If $k=0$, meaning that there is no obstacle in the flight path, the equation 71 above gives a result of $i2$. $i2$ holds only the 0 intention meaning level flight as seen on step 1 model above. When $k=1$, meaning the existence of the obstacle is notified by the neural fields, i.e. activity detectors, equation 71 gives the output $i1$. $i1$ is simply position of the obstacle added to the lateral size of the aircraft with a limited safety separation. This is shown on equation 71. On equation 71; used parameters are shown below.

D: Obstacle avoidance azimuth intention value

X_{obs} : Azimuth position of the aircraft

X_{ac} : Aircraft wing span

∂ : Safety margin.

Finally the output of detection switch mechanism shall be memorized on a 1D neural field and shall be used in all pilot model architecture. Notice the above architecture is planned to make a lateral movement to pass the obstacle and not to change the pitch attitude of the aircraft. So pitch attitude is used as given in step 1 model. Unified step 2 model is given on below figure.

Improved pilot model given in step 2 is expected to make a level flight similar to real pilot in case of free air field, and perform a manoeuvre to avoid the obstacle in case of it exist in the field of view. The results and discussion will be given on later on paragraphs.

4.4 Step 3: Pilot Model with Altitude Effects on Hold Performance

In the improved pilot model, pitch control is performed using still the step 1 constant intention pilot model. This model given in step 1, shows the same elevation hold performance and pitch control without considering the altitude and always holds the aircraft level even an obstacle exists. All obstacles are avoided only by a lateral manoeuvre. But in fact, this is not the real case. Our research on vertical hold performance of the real pilots on real flight test data shows that in lower altitudes close to

ground, pilots shows a better hold performance with lower elevation deviation and performance or handling quality of the vertical attitude is quite diminished with increasing altitude. Those results are given in later chapters. These performance changes significantly related to altitude are probably results of various neurotransmitters' density changes within the effects of emotional aspects such as fear felt while flying on close proximity to ground or the pilot being trained to be more careful under such conditions. In any way, being connected to the emotions, level of consciousness or training, flight altitude is observed to affect the performance of the pilot in real flight tests. These effects are shown with a yellow block in figure 33.

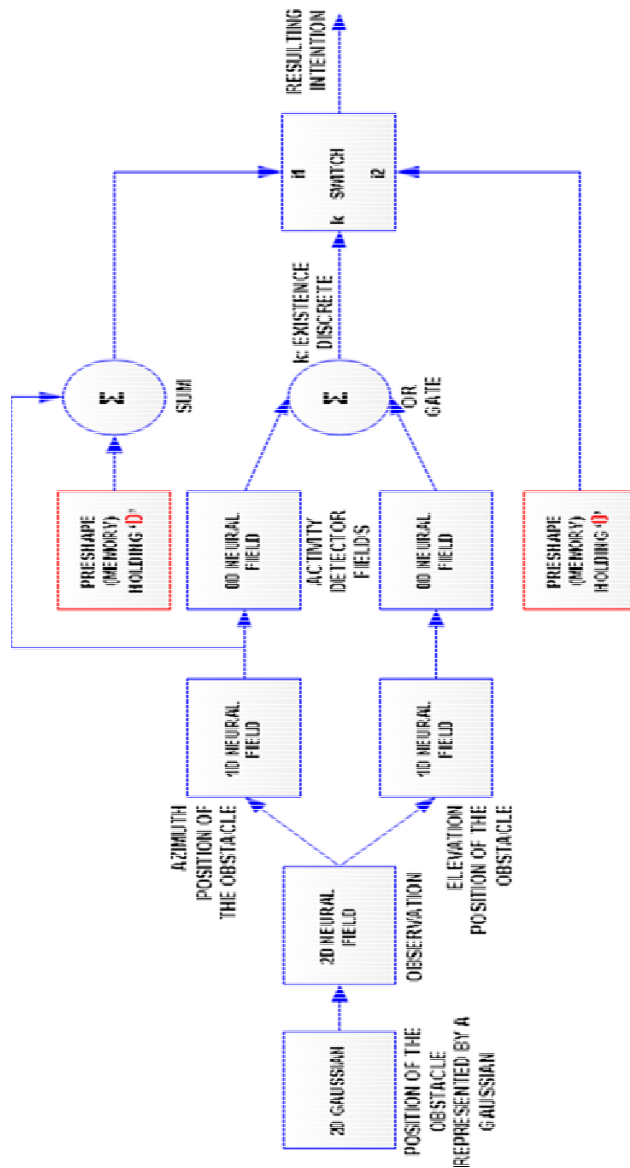


Figure 31. Detection Switch Mechanism Representation. The circuit is used to make a selection between the cases where there is an obstacle in the centre of the screen or not. In any case, the switch function makes a choice regarding the value of k. The “factor” k can only be 0 or 1 regarding the outputs of the 0D neural fields.

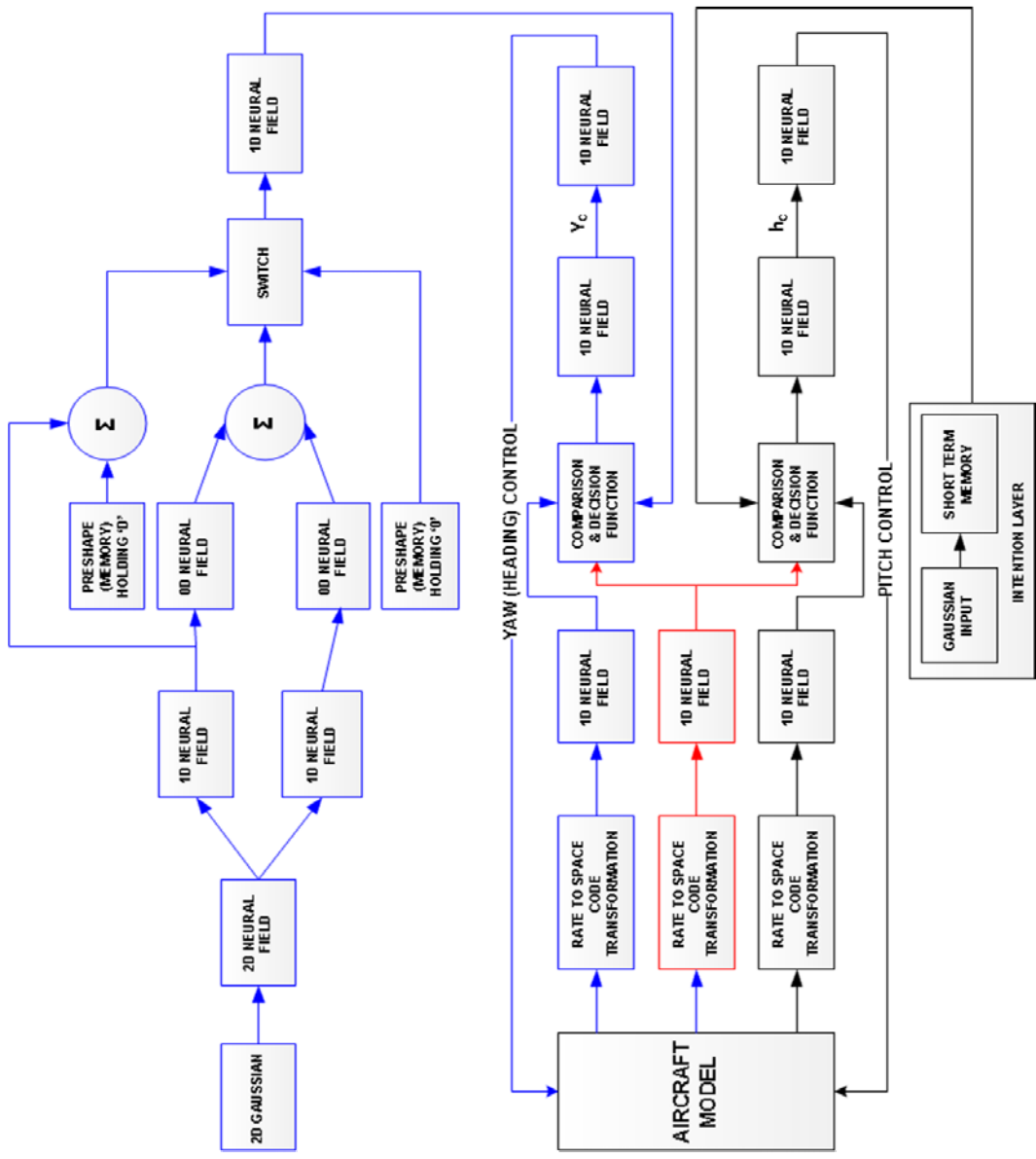


Figure 32. Overall Improved Pilot Model Representation. The lateral hold intention is modified with the existence of an obstacle.

Such a function can be added to the pitch control channel of the improved pilot model. Pitch or elevation hold performance can be adjusted with a neurotransmitter density adjustment function using the altitude. The details will be given on later on paragraphs. But the main point is that the effect of the “emotions and training” or in other words altitude effects are simply added to the pitch channel to create a pitch control complexly proportional to altitude. On below figure the emotionally modified pilot model is given.

Since the DFT approach claims on being embodied but not biologically mapped into any cortex, the connection of the yellow box into the pilot model may vary due to the modeller’s approach. In this study the method implemented will be described on later paragraphs.

Notice the effect of the altitude on the hold performance may vary from pilot to pilot regarding the experience, cognitive state of the pilot including the effect of emotions and from flight to flight regarding the length of mission and level of hardness. Therefore it is not easy to construct a statistical model based on the flight test results exactly covering a wide range of performance but it is still possible to create some amount of correlation.

For further discussion on the architectural issues on yellow block please refer to paragraph 5.2.2.

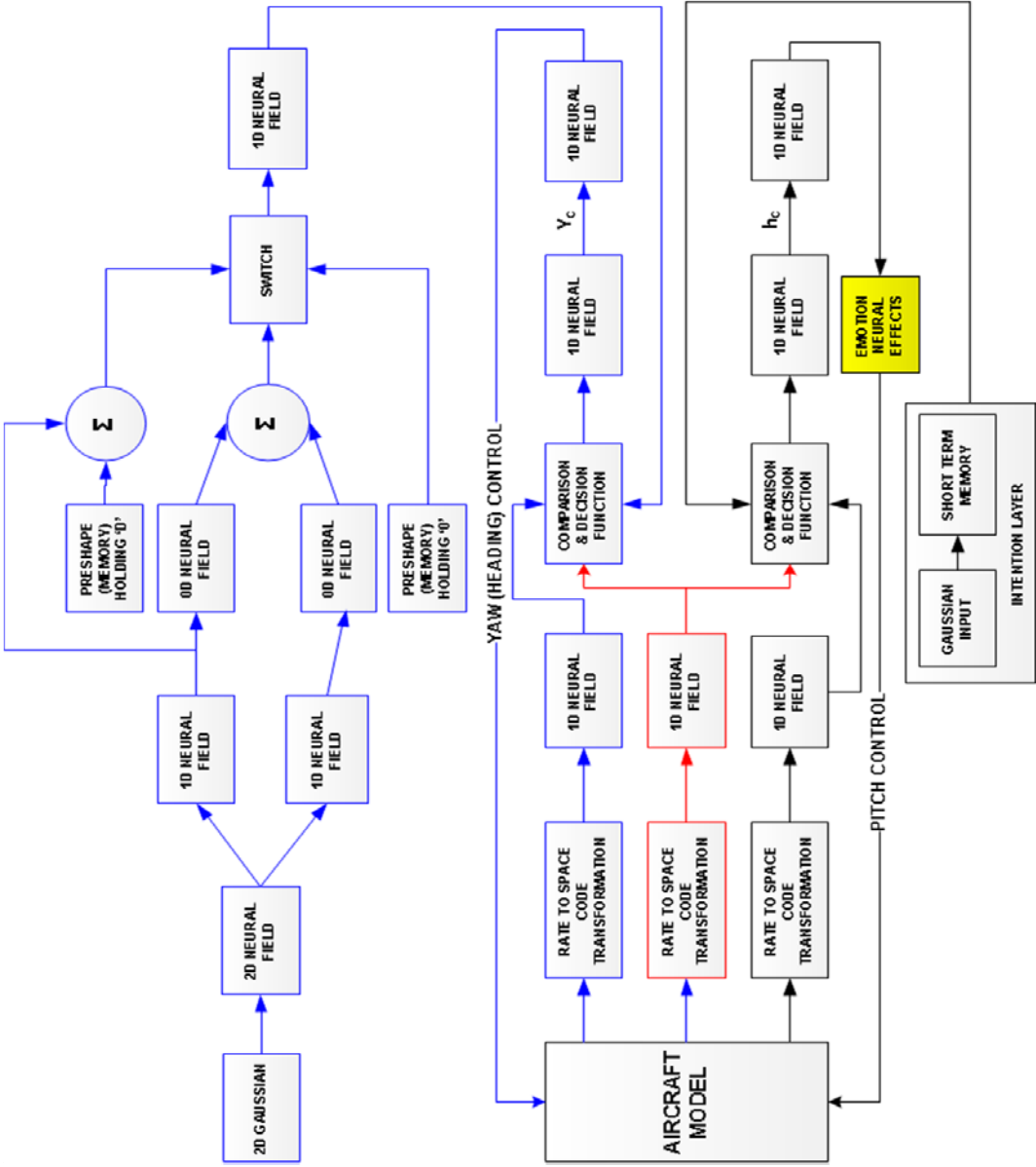


Figure 33. Pilot Model with Altitude Effects on Performance. The effects of altitude is added to the pitch channel.

Statistical comparison between the real flight test data with human pilots and the step 3 model will also be given on results chapter.

Further details on this subject are given in paragraph 5.1.1 including the human pilot performance data and 5.2.2 including architectural evaluation.

4.5 Negative Deviations' Representation in The Spatiotemporal Axes

The architecture given in figure 30 and 31 are in fact simplified for sake of understanding. Normally DFT approach does not allow the fields to represent negative spatiotemporal axis values directly. The only ways to represent negative directions (such as downward deviation in a positive up axis setup, or left in a positive right setup) can be indirect methods. In this study such an indirect method is developed to overcome such weakness of DFT method implemented in CEDAR. This issue is in fact irrelevant with the pilot cognition but it is directly arising from a weakness of DFT. So it is not in fact a part of the embodied model but in fact just a modelling trick to overcome a difficulty. The method is summarized in below figure. Each representation of spatial value is divided into two different fields each representing different directions. So the architecture in fact contains 4 different channels different than the 2 simplified demonstration channels given in above figures. For further information on axis construction please refer to paragraph 6.1 on discussion on weaknesses of DFT.

Notice after all of the decision blocks in figure 24, a motor block was given. When the figure 32 is observed, it will easily be noticed that there are two different control lines ending with a summer. This summation may rise the question of that what if the resulting control amount of these two fields may cancel each other due to the difference of control directions. At that point, it shall be noted that the rate to space code transformation for two different directions may never give rise to an activity at the same simulation step since the aircraft hold deviation will be negative or positive at a single moment but never both at the same time. Therefore, if the negative or positive rate to space code transformations will never be active at the same simulation step, than the resulting control outputs given in figure 32 will never be active at the same moment. In a single moment only one control direction can be active in that architecture.

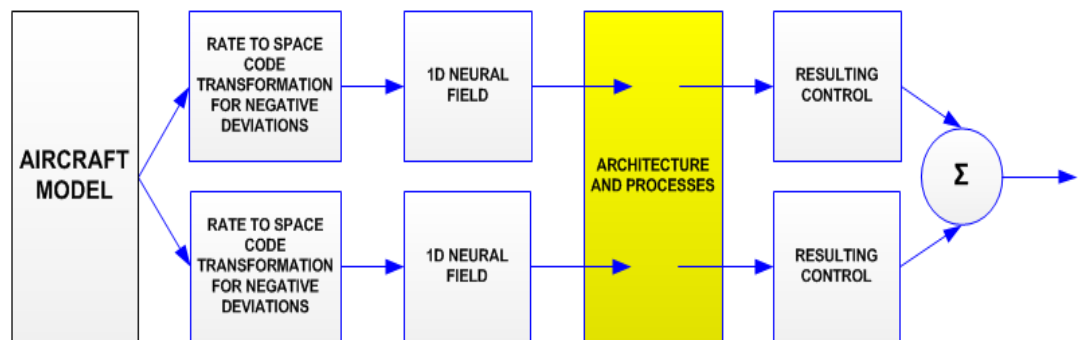


Figure 34. The method of dividing a spatial variable into two fields for representation of different directions in an axis.

One important fact is that the amount of the control generated shall be above a threshold to be able to create an activity in the control output field due to the inhibitory kernels included in control fields to eliminate noisy signal effects. Another important parameter in the control fields is the time constant of the field activity which gives rise to possibility of a muscular time delay. In fact muscular delay's usage or addition to a pilot model is not a new approach and it is also included in some of the control models reported in paragraph 2.1. On that point DFT provides a great advantage since the delay is not

created with a transfer function but it is an embedded property of the neural fields created by a natural property and such property is completely different than the control theory based modelling approaches.

4.6 Network Elements

Although they are not shown on above architectural drawings, in fact the DFT based pilot model architecture contains some network elements to make fine tuning or signal matching between building blocks. On the paragraphs below some simple explanation on embodiment meaning of such network elements are given.

One of most common network elements is “**static gain**” element. These elements are simple multiplication elements using in the network when the output of a field or any other object is needed to increment, decrement or sign change for a value is required. Static gain element performs below mathematical function regarding what is the gain factor.

$$f_2 = f_1 * k \quad k=-1 \parallel 0 < k < \infty \quad (\text{Equation 72})$$

Where;

f_2 : output of the static gain function, can be a scalar or a matrix regarding what f_1 is,

f_1 : input to the static gain function, can be a scalar or a matrix,

k : static gain value, which can be -1 for a sign change, (0, 1) for division and $1 < k < \infty$ for signal incrementing.

Static gain elements can be assumed as representations of biological fields with linear kernel functions where threshold is set at 0 and there is no inhibitory kernel.

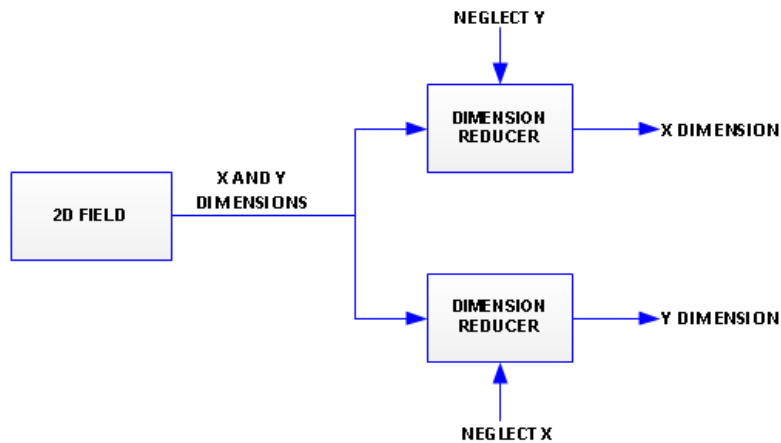


Figure 35. Dimension reduction function.

Another common network element is “**dimension reducing**” or “dimension resize”. Dimension reducer blocks are used in network when a higher dimensional field output is needed to decrease to a lower dimensional field. E.g. a 2D neural field output is needed to be analysed separately for each dimension. A dimension reducing block can be used for each of dimensions. Please refer to above figure 34 for understanding the proper function.

As it can be seen on above figure 35, dimension reduction is a very useful network element since many fields separation under subcomponents is made via this block. Despite its mathematical function in matrix level, this block is in fact can be used to minimise the sizes of single dimension fields (e.g. as from 100 to 80). Therefore it can be assumed being biologically equivalent of a 2D field with linear kernel with threshold set at 0 in one dimension and completely inhibitive kernel in the other one. So it is inhibiting one dimension of a field input and directly transferring the other one.

A similar network element used very often is “**summer**”. As it is clearly understood from its name, a summer gives the addition operation result for given inputs set not limited with two members. A summer is representation of a neural field with no kernel which transfer all inputs to output without gain.

Network elements are not shown in above step 1 to 3 architectures due to the simplicity purposes.

This page is intentionally left blank.

CHAPTER 5

5 RESULTS, DISCUSSION AND CONCLUSION

5.1 Results

5.1.1 Understanding the Human Pilot Vertical Hold Performance

Human pilots, similar to a mathematical model, also may exhibit performance degradation during vertical and lateral navigation performance. To be able to evaluate the performance of the model given in this study, human pilot performance needs to be known and understood. When an autopilot system is not on duty, generally every aircraft shows slight deviations from planned path of navigation. These deviations may be caused by the aircraft dynamics, weather conditions and also from human control performance. In addition to this, the pilot may allow the aircraft slide out or deviate from planned route or course unconsciously due to changes in cognitive workload in the cockpit caused by the presence of multiple tasks that require the pilot's attention, or due to the motor ability level of the pilot.

In this study, the primary focus was on the pitch hold performance in level flight since it is relatively easier to understand and analyse. In order to assess human piloting performance, data from previous flight tests are analysed. These flight tests are originally not conducted for this study and in fact each flight's task was different. So in fact the pilots were not focused on creating a level flight hold performance since this is not the primary task to perform. But still the pilots were mandated on performing a level flight in a part of the overall test procedure. The identities of the pilots are not known, so performance assessments could not be made at the individual level. The corpus of human piloting data obtained from the flight tests is assumed to be representative of generic human piloting behaviour.

The tests are conducted with a T-38M aircraft, a jet trainer (modified Talon) with a mechanical flight control system; and electronic flight controls, autopilots, flight directors and management systems, assistance or stability systems are not used. The lack of such systems results in that the measured performance is completely a function of aircraft

dynamics and pilot cognition. Even the wind vector data on ground level is known, but since the altitude winds may possibly be different than ground winds in Turkish climate, flight test environmental conditions are neglected.

Table 1 below shows the vertical hold performance results of level flight tasks for 15 different flights at different altitudes. Notice the deviations may be negative or positive from flight level. When an analysis of such performance is desired, absolute values of these values may be useful.

Table 1. Human Pilot Unfocused Vertical Hold Performance.

Test ID	Target Altitude (AGL) (ft)	Mean Error From Target (ft)	Maximum Positive Error (ft)	Maximum Negative Error (ft)	Standard Deviation	Absolute Value of Average Error (ft)
1	2750,00	0,00	35,00	-20,00	14,44	0
2	2850,00	-8,46	10,00	-25,00	7,36	8,46
3	4878,00	-0,29	25,00	-20,00	10,64	0,29
4	5428,00	7,17	80,00	-35,00	28,61	7,17
5	5878,00	-24,05	50,00	-55,00	17,21	24,05
6	6398,00	9,14	25,00	-25,00	12,03	9,14
7	6950,00	11,61	72,50	-55,00	5,26	11,61
8	7300,00	-25,68	10,00	-40,00	13,01	25,68
9	7900,00	-10,96	40,00	-35,00	20,10	10,96
10	8000,00	-10,40	45,00	-57,50	23,38	10,4
11	8768,00	2,51	10,00	-5,00	3,87	2,51
12	8978,00	24,59	102,50	-175,00	78,20	24,59
13	18150,00	11,92	35,00	-5,00	11,36	11,92
14	18200,00	23,69	70,00	-22,50	22,73	23,69
15	19750,00	-15,59	45,00	-82,50	29,44	15,59

The very first impression given by table 1 is that pilots' performance tends towards making positive (upwards) deviations rather than negative deviations, probably for omitting a ground collision. 6 flights out of 15 include negative mean error values and 9 include positive mean deviations. Here the word "error" is used to represent the deviation from target altitude to avoid confusion with the statistical concept of standard "deviation". Similarly there are only 6 altitudes where the absolute value maximum negative error is larger than the maximum positive error. While 7 of the flight data are below 7000 ft altitude above ground, 5 of them contains mean absolute errors below 10 ft. Above 7000 ft there is only 1 flight containing a mean absolute error below 10 ft. Table 1 points out that the pilot vertical hold performance shows a decrease in error when the flight is on close proximity to the earth's surface. Therefore, it can be deduced that the vertical hold performance of the human pilot can be a function of height above ground. Of course the function may include other inputs such as flight duration (i.e. is the pilot

tired?) or weather conditions, but under current circumstances of this study, these inputs are not available and analysed.

The mean value of mean absolute errors is 12.4040 ft. showing that for an unfocused level flight independent of the altitude, it is expectable that the aircraft altitude may differ a mean 12 ft from level attitude. Mean of maximum positive errors is about 43,67 ft and mean of maximum negative errors is about -43.83 ft showing that the maximum errors on both upward and downward directions may be expected to be similar. Table 2 summarizes the descriptive statistics of the data given in Table 1.

Table 2. Descriptive Statistics for Flight Test Results.

	N	Minimum	Maximum	Mean		Std. Deviation	Variance Statistic
				Statistic	Std. Error		
Absolute Value of Mean Error (ft)	15	0,00	25,68	12,4040	2,24348	75,498	75,498
Mean Error From Target (ft)	15	-25,68	24,59	-,3200	4,00198	240,238	240,238
Maximum Positive Error (ft)	15	10,0	102,5	43,667	7,1083	757,917	757,917
Maximum Negative Error (ft)	15	-175,0	-5,0	-43,833	10,8284	1758,810	1758,810

Table 3. Skewness and Kurtosis Statistics for Flight Test Results.

	Skewness		Kurtosis	
	Statistic	Std. Error	Statistic	Std. Error
Absolute Value of Mean Error (ft)	0,288	0,580	-0,998	1,121
Mean Error From Target (ft)	-0,042	0,580	-0,753	1,121
Maximum Positive Error (ft)	0,653	0,580	-0,133	1,121
Maximum Negative Error (ft)	-2,404	0,580	7,029	1,121

Regarding the above table 3, the frequencies of the vertical errors in the flight test results do not show a normal distribution. This result can be observed in Figure 36. Regarding this non-normal behaviour, an ordinal organisation cannot be constructed between the occurrence frequencies of mean errors.

But what is the relation between the altitude and the performance? Can it be statistically meaningful? First of all, there is a low-to-medium level positive correlation between these two data sets (0.393). Below table 4 shows the correlation between altitude and the vertical hold performance.

Although there are other effectors, it is clearly seen that there is a relation between the pilot's vertical hold performance and the height above ground at a level of medium correlation at least during level flight. This relationship can be used for the modelling of

the emotional and training effects on a pilot model. To be able to provide a mathematical function of these effects a statistical regression model is used in this study.

When an initial boxplot analysis of the test data is conducted it has been observed that none of the mean error values are outliers. Despite the tests with ID 5, 8 and 12 shows extreme error means in comparison with the similar altitude tests, still they lie inside 1.5 median boxes. So for the very first analysis they are not eliminated. Since the pilots' identities and experience levels are unknown, there is a probability that these 3 tests belong to an unexperienced pilot or to the final period of a long and tiring test where the pilot in duty is tired. This may be the reason behind the results of these 3 tests are visually observed as being different from similar altitude tests.

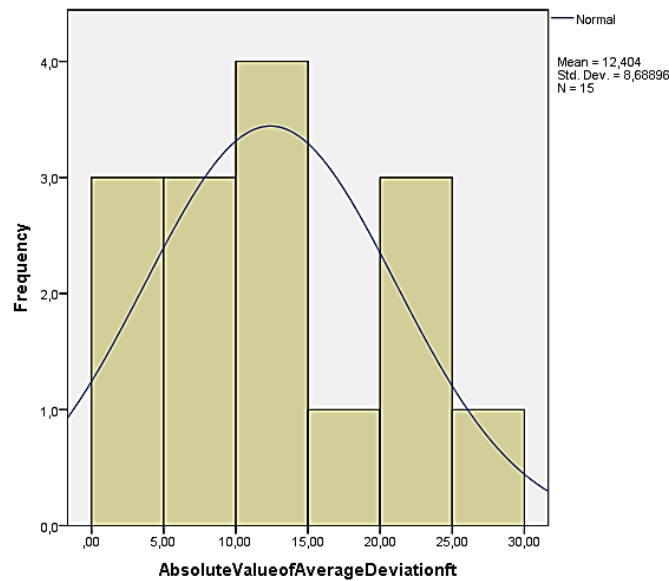


Figure 36. Histogram of absolute Values of Mean Errors.

Table 4. Pearson Correlations Between Target Altitude and Absolute Mean Deviation.

		Target Altitude AGL ft
Absolute Value of Mean Deviation (ft)	Pearson Correlation	0,393
	Sig. (2-tailed)	0,147
	Sum of Squares and Cross-products	260874,788
	Covariance	18633,913
	N	15

Two different regression models are conducted for this first data set including 15 flights one being linear and one being logarithmic with respective R^2 values of 0,219 and 0,155. Notice these two results may be accepted as very bad fits. Especially the linear fit does really not provide a statistical correspondence with given test results. Figure 37 below shows the results of both full data set and eliminated data set regression results.

As previously mentioned the tests 5, 8 and 12 show a different behaviour in comparison with the other tests conducted between 5000 and 10000 ft. The average of these 3 tests' mean error values is 24,77 ft being 287% times greater than the average of other 6 tests in the same interval being 8,63 ft. Regarding this fact, the data for these 3 tests were eliminated to observe how the regression results will be affected. We name this data set as the eliminated data set and it is given in the table below.

After the elimination of these 3 cases, a new regression analysis produced a much better fit to the remaining data points. Logarithmic, quadratic and linear regressions are conducted and while all providing R^2 values above 0.5 (medial fit) linear regression model provided 0.548 showing best between all 3. The results are shown in below figure 36.

Table 5. Eliminated test data set.

Test ID	Target Altitude (AGL) (ft)	Mean Error From Target (ft)	Maximum Positive Error (ft)	Maximum Negative Error (ft)	Standard Deviation	Absolute Value of Average Error (ft)
1	2750,00	0,00	35,00	-20,00	14,44	0
2	2850,00	-8,46	10,00	-25,00	7,36	8,46
3	4878,00	-0,29	25,00	-20,00	10,64	0,29
4	5428,00	7,17	80,00	-35,00	28,61	7,17
6	6398,00	9,14	25,00	-25,00	12,03	9,14
7	6950,00	11,61	72,50	-55,00	5,26	11,61
9	7900,00	-10,96	40,00	-35,00	20,10	10,96
10	8000,00	-10,40	45,00	-57,50	23,38	10,4
11	8768,00	2,51	10,00	-5,00	3,87	2,51
13	18150,00	11,92	35,00	-5,00	11,36	11,92
14	18200,00	23,69	70,00	-22,50	22,73	23,69
15	19750,00	-15,59	45,00	-82,50	29,44	15,59

As it is observed in above description and below figure 36, a linear regression with eliminated data set may provide a mathematical representation between the altitude and the human pilot vertical hold performance. Under this circumstances, below equation 7 can be used as a figure such relation.

$$C = 0.0008 \text{ HAG} + 1.896 \quad (\text{Equation 73})$$

On above equation the terms are;

C: Mean of maximum error in vertical hold performance,
HAG: Altitude in ft, as height above ground.

The results of using such equation will be described in later paragraphs and a discussion on methodology will be given.

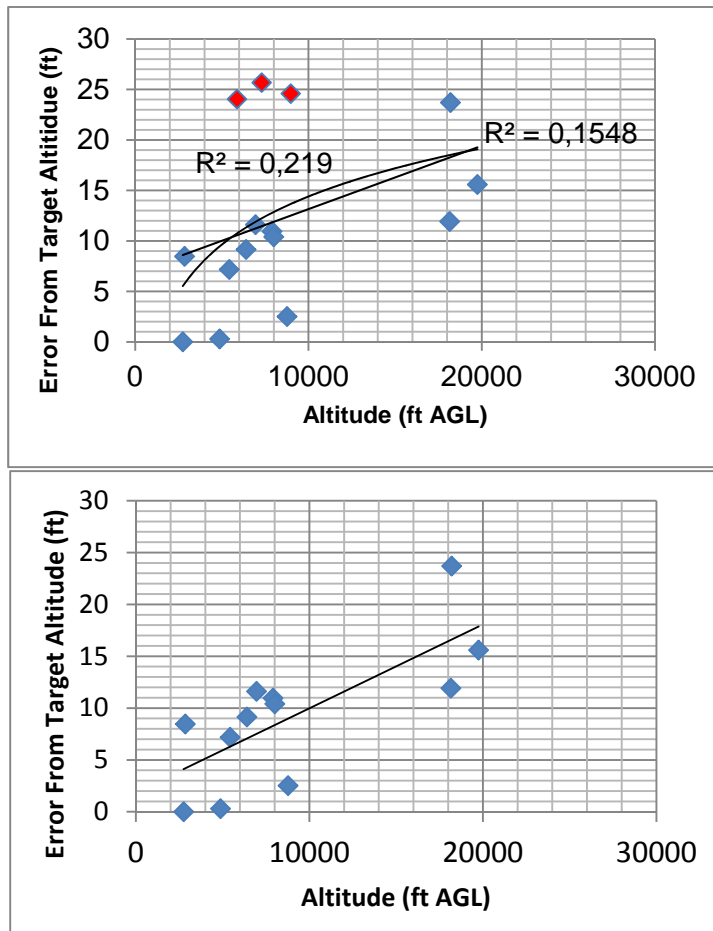


Figure 37. Regression models: with full data set (up) and eliminated data set (down). The red data points in left chart are eliminated in right one. Best regression line (linear) shown with red colour in right.

5.1.2 Model Outputs

To be able to compare with human pilot performance, the model performance shall be evaluated. For this purpose multiple flights with different parameters at different altitudes (height above ground, ft) has been performed. The model has been evaluated under varying status of obstacles existence and the emotional and training aspects that has been mentioned on previous paragraphs has been observed. For the applied modification between step 2 and step 3 models please refer to paragraph 5.2.2. On below paragraphs model flight results will be provided.

First of all pilot model, even the step 3 improved one has crashed the aircraft in all flights below 2000 ft. This is a result of extreme complexity of the model and difficulty on fine tuning of the model parameters due to mentioned complexity. Since the model includes more than 20 blocks of neural field and various network elements, an optimum performance at low altitudes requires much more study of model parameters' optimisation. But the situation changes above 5000 ft.

On all altitudes including low altitudes, the models mean of down and up deviations is quite close to 0. This situation only slightly improves for step 3 improved model flights. On below chart the mean deviation numbers are given for both step 2 and step 3 models.

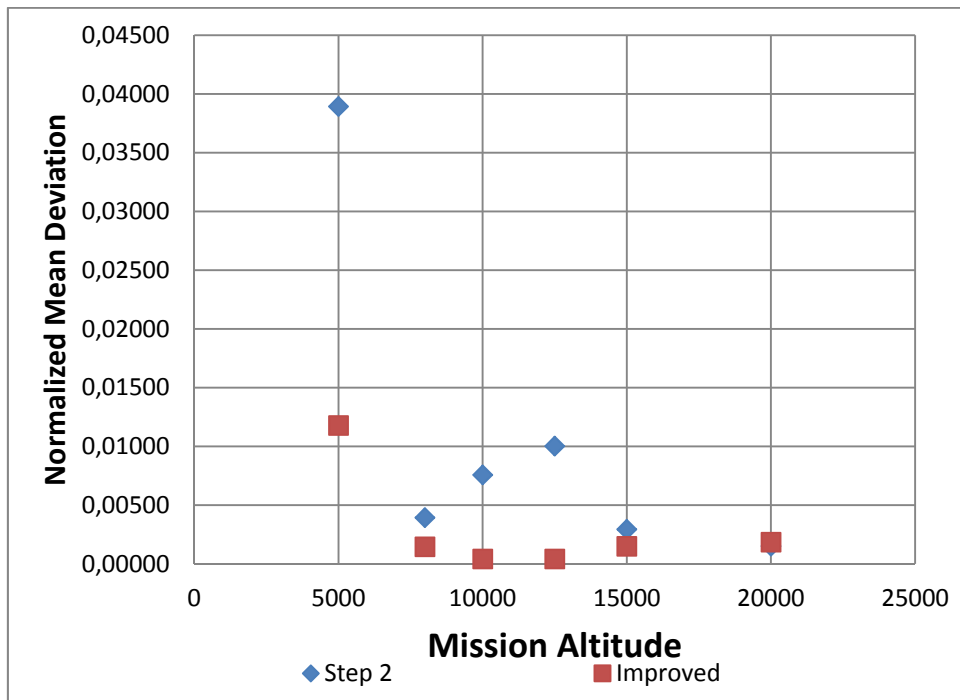


Chart 1. Mean deviation performance for model pilot. Normalized values represented. Results of 30 flights at varying altitudes and with varying parameters. Altitudes below 5000 ft neglected due to crashes.

As it can be seen from above Chart 1, step 3 model mean deviation is very low, i.e. below %5 of mission target altitude for except one case at 5000 ft. Similar quality is valid also for step 2 model flights. Notice at low altitudes the model has an aim towards showing relatively larger mean deviation values due to the capability lack of generating further smaller controls. Especially for step 2 model, it is clearly seen that the model is less successful at low altitudes.

Despite the acceptable mean hold performance shown above, the ground crash protection is a matter of eliminating maximum deviations. Below chart provides a representation of maximum deviations from target altitude for both models.

Both models step 2 and step 3 makes the maximum deviation at 5000 ft as mentioned on previous paragraphs. Step 3 model improvement results in a maximum deviation of %24.803 at 5000 ft HAG while step 2 gives a maximum deviation of %137.669 pointing out a crash at this selected altitude. Please see below chart. But for altitudes, the mean of maximum deviations is %39,29 for step 2 model and %0,959 for step 3 model. Notice under average conditions, step 2 model is also not crashing, but at low altitudes, a crash is possible and observed for all altitudes below 2000 ft. Another important point is that there is a strong correlation between the two data set belonging to two different models' maximum deviation means ($r^2 = 0,7862$) showing an improvement on performance under

the effect of the additional blocks simulating the emotional and training effects. i.e. the knowledge or perception of the current altitude creates a positive improvement on the model performance. This situation can be observed on chart 3's trend lines as an offset difference.

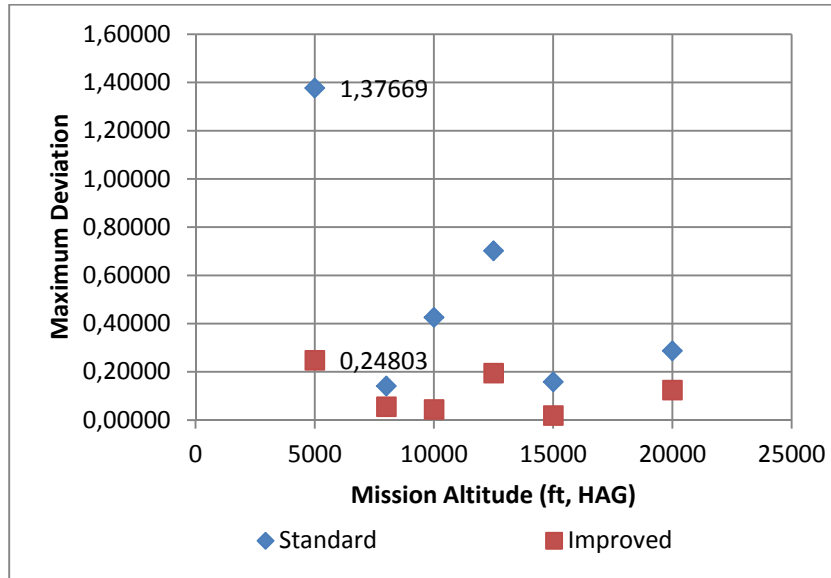


Chart 2. Mean maximum deviation performance for both step 2 and 3 represented.

Here there rises another question: can there be a numerical merit of the improvement added by the step 3 valid for all altitudes? Since the models create different performances at different altitudes, a valid quantitative merit of improvement cannot be simple the means alone, although they provide some meaningful information. Here it is preferred to see if a regression can be done using a linear model. In above chart 3 the results of these regressions are shown. First of all, R^2 values for both models are quite bad, 0.0017 for model step 2 and 0.07 for model step 3. It can be deduced using very bad R^2 value of model step 2 that, simply a model which is not aware of the altitude does not change its performance with altitude. But when the observation is focused on model step 3 performance, it is seen that the regression quality is much larger even it is not a good fit. The linear regression results have the form given in below equation 74:

$$\text{Mean Maximum Deviation} = \text{Multiplier} * \text{HAG} + \text{offset} \quad (\text{Equation 74})$$

The offset values for models are 0.3691 for step 2 and 0.0338 for step 3 models respectively. Clearly regarding the offsets, the step 3 model creates an improvement in the model performance and regarding the R^2 value (0.07), results in a slightly significant ($R^2 > 0.05$) altitude dependent performance. As we have stated previously and as observed in human pilot, the performance was also not expected to be a single function of altitude. Additionally a finer tuning of model parameter could result in a further strong dependence.

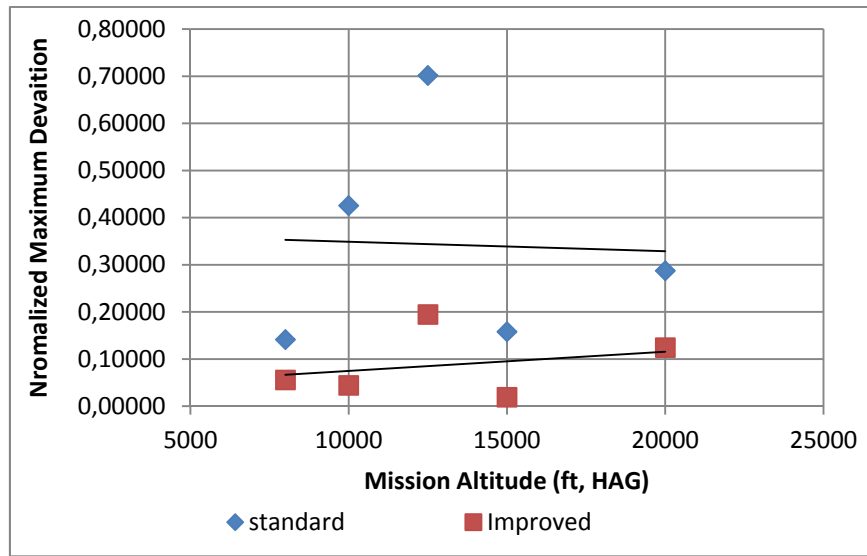


Chart 3. Model data sets' comparison above 5000 ft HAG.

Finally how does the altitude changes look like during a complete flight should also be described here. Please see below chart for a better understanding showing a 10 minutes flight.

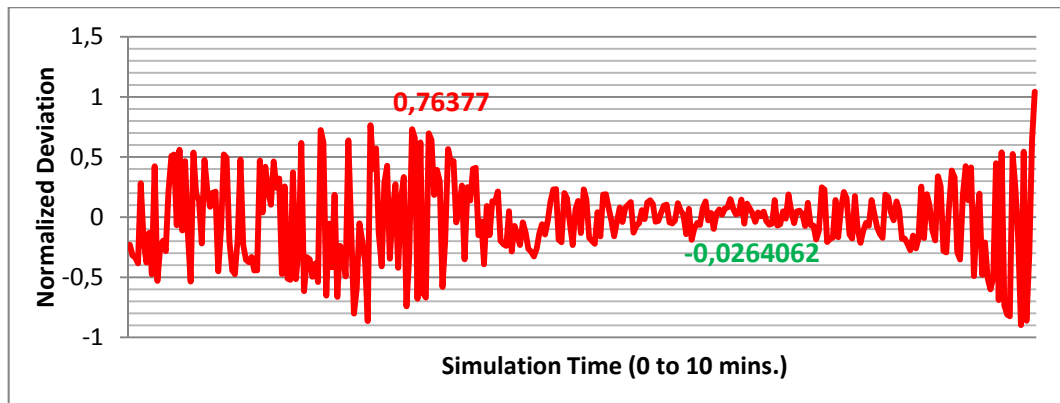


Chart 4. Vertical position hold performance represented during a whole mission flight (10 minutes).

In above chart, the aircraft is continuously going up and down above target altitude, but the man altitude is very close to the target one. In this flight, the first part is performed by disconnecting the model 3 update and using only the model 2 performance. As expected the model makes large errors up to %76,4 of the target altitude. After about the fifth minute of the simulation, the step 3 improvement connection is made and the performance of the model is clearly improved. The errors are generally fallen down to an amount below %10 at that part of the flight. Finally at a point after 8th minute of the flight, the step 3 connection is again broken by hand, and the model started to fly the aircraft in an increasing error until the flight ends with a crash.

5.1.3 Comparison Of Model and Pilot Performances

Until that point, the performance of the overall model and ant the effects of the improvements are summarized. It is shown that the addition of an awareness of altitude as a representation of emotions and training effects on pilots, creates a further improved hold performance at high altitudes and is at least better than the relatively simpler models in low altitudes. Now let's see if the model shows similarities with human pilots.

First of all comparing the means of maximum deviations, human pilot performs her/his function much better than the model pilot. Please see below chart for a comparison of model and human performances.

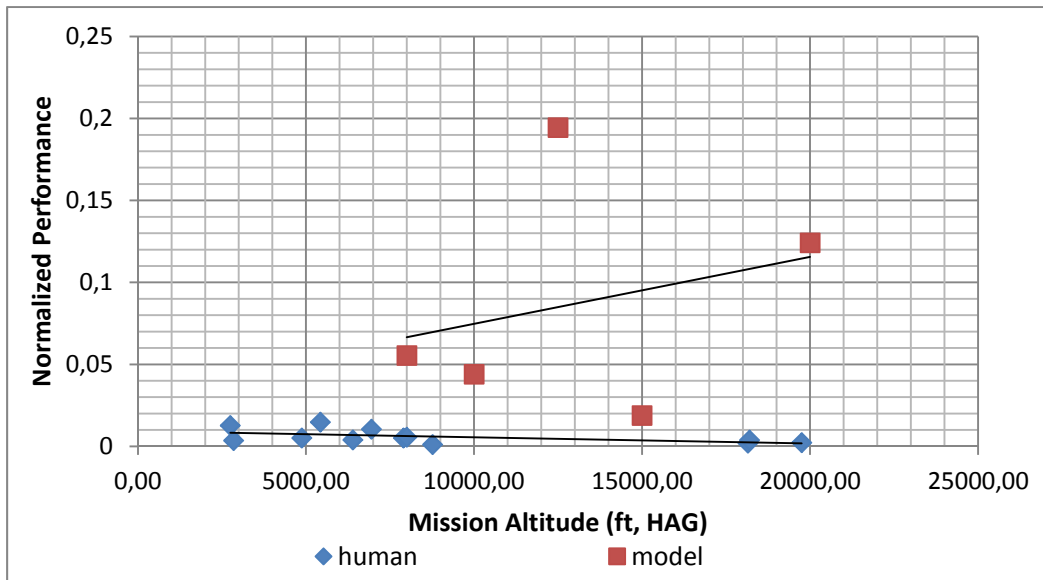


Chart 5. Human and model pilot data sets at different altitudes represented.

First of all human performance on vertical position hold is about %0.5 and this value for even step 3 model is %0.9. The performance difference is about twice and it shows that the model pilot does not reach the performance of human pilot. **But regarding the overall flight results in a normalized way, the correlation between human and pilot model data is acceptable ($r=0.3435$) for flights above 5000 ft.** For lower altitudes, there is not a significant correlation due to the various crashes making a computation of mean maximum error impossible.

Here one point is very important: The human pilot errors grow with increasing altitude probably as an effect of the safety provided by high altitudes. But the dependency between the increase in error is not as fast as the increase in altitude. In other words, the errors are higher at high altitudes but the relation is not linear. This behaviour can easily be seen in chart 5, a linear trend line for human pilot data is shown and as described the normalized errors trend line (a linear regression model) is decreasing with a very straight slope of $-4.0 \cdot 10^{-7}$. But for model step 3, the pilot performance is decreasing with increasing altitude and shows a slowly increasing trend ($m=4.0 \cdot 10^{-6}$). Therefore it can be said that the DFT model pilot is much more dependent to the altitude changes. Here it

shall be noticed that the model pilot trend is just slightly representing the data set ($R^2 = 0.07$).

Similarly the offset values of these linear regression models, also points out a difference between the performances of the model and the human agent. The human agent performance regression offset is 0.0093 and the model step 3 pilots' one is 0.0338 showing a standard maximum error difference about %2.45 of mission altitude for all flights. In other words, for a flight at 10000 ft, the model pilot may be accepted to make a 245 ft larger mean error in comparison with a real pilot. For a further improved performance matching, the model parameters shall be adjusted better. The possibility of such an improvement will be discussed in later paragraphs. Additionally a far better performance can be obtained with an adaptive or learning model.

5.2 Discussion and Conclusion

5.2.1 Weakness of DFT Approach on Negative Axis Description

The model described in this study contains the perception of a spatiotemporal variable and a decision regarding the value of that subjective variable. The Amari equation can be conveniently used to “represent” this variable, but it shall be noted that the activity value u of the neural field can only be positive when it exists, or else it will decay to the resting level h . Clearly the activity level of the field can be a representation of any spatiotemporal value but it can represent the existence or notification of such percepts. So the quantitative value of a spatiotemporal variable (such as position of an object) needs to be represented by the position of the activity at a spatiotemporal axis. To be able to implement such an architecture, the developer or modeller shall construct an axis setup to provide the acquisition of the variable.

Current DFT modelling frameworks like CEDAR do not allow the user to directly implement a spatiotemporal axis with negative values. The axis needs to start from 0 and shall end in any value greater than 0. In other words, the position of an object shall be represented between, if angularly represented, let us say from 0 to some value such as 220 degrees where 0 representing one end of the visual periphery and 220 the other far end. In such a case of axis construction, the mid-point of the parameter space being 110 degrees shall represent the centre of the visual field. A similar approach can be applied to vertical organisation of the perceptual or decisive fields.

Notice this structure cannot be proven to be completely compliant with human cognitive fields. A human can perceive the objects on the left or right and down or up, and being up does not imply a value more positive than being down. In other words a positive activity in a neural field may represent easily an opposite direction with equal magnitude with just a change of orientation. So in fact human cognition may be accepted, at least for the sake of simplicity for modelling purposes, to contain sets of axes varying from any negative value, crossing 0 (or any reference point for the centre), and ending with a positive value. In this way, positive activities may also be allowed to rise up on the negative axis values.

As previously mentioned, existing DFT tools do not directly allow such a construction. This important issue creates an important structural difficulty on the development of the pilot model discussed in this study. Notice that the pilot model shall create a downward control input to the aircraft when the aircraft deviates from the target in upward direction and vice versa is valid when downward deviation occurs. Under such circumstances, the

spatiotemporal axis of the aircraft's lateral (horizontal) position shall be divided into two different fields to be able to organise two different directions. This situation is summarized in the following figure.

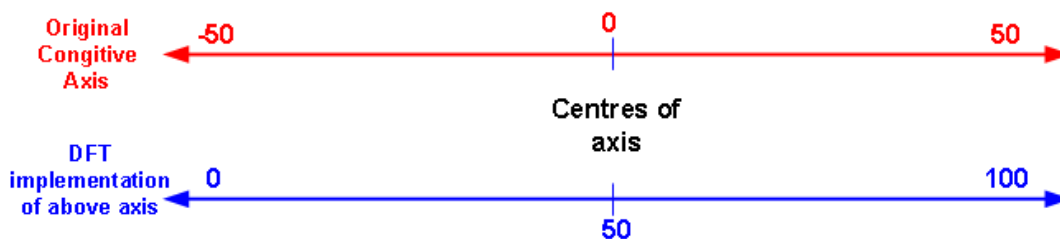


Figure 38. The relation between the DFT implementation of a spatiotemporal axis and original representation is shown

Let us assume the spatiotemporal axes are divided into lateral and vertical dimensions by the dimension reduction tool of CEDAR and leftward point in the lateral axis is selected as 0, resulting in the far right end being the maximum value of the axis. When the aircraft lateral deviation is observed by a space to rate code transformation, if the observed value is negative, it may be mapped into a field processing the perception operation for leftward deviations, and if positive the mapping shall be oriented towards another field for rightward perception. This situation is previously shown in section 4.5 with figure 34. The method of dividing each axis into two different field representations is used in this study to overcome the problem of non-existence of negative axis definition in CEDAR. When we apply this method to the given simplified architecture in section 4.3 the architecture becomes far more complicated in comparison with given in figure 29 since each channel given in the figure is in fact divided into two separate channels representing two different directions of axis ends.

5.2.2 Weakness of CEDAR DFT Framework on Adaptive Model Parameters

The statistical evaluation of the human pilot performance and the corresponding simple mathematical model was given in section 5.1.1. Notice that there can be various methods of adding such effects in a simulation. One very first idea on implementation can be to add a block of altitude evaluation (the yellow box in section 4.4) as given in step 3 model architecture. This block may be assumed to represent the function of emotional cortical parts in the human brain unified with the long-term memory, which is assumed here to contain training effects. This block could also be connected to control the decision blocks, the level of perception alertness and sensibility in motor cortices. Biologically this type of function can be similar to the effect of amygdala for the release of neurotransmitters providing a change on electrical behaviour of affected cortices or nuclei. In this study a method similar to this approach is used. The implementation contains a switch mechanism under the control of altitude to modify the decision field blocks' activity outputs. The mechanical switch approach is summarized in the following figure.

Above mentioned method of complex switch function in fact is not biologically based. In a biologically based model, the altitude and training effects block should in fact change the relevant cortices' time constants, kernels and threshold functions to shape the output field activity. Here, in the mechanical switch approach, normally the decision fields' activity is not affected. But despite that necessity, only the current output activity is

mathematically modified under the effect of the altitude mechanical switch in a compatible way with the mathematical equation given in section 5.1.1. This is not different than a summing operation. In fact CEDAR does not allow the time constant or the kernel of a neural field to change adaptively during the simulation. Even worse, the parameters of each field need to be manually manipulated while the simulation is running. So it can easily be deduced that such weakness creates a difficulty for constructing such adaptive models where emotional or task effort aspects are implemented.

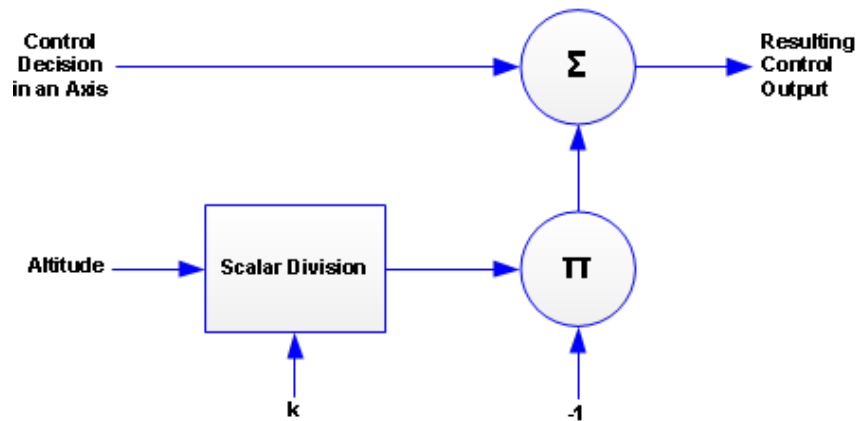


Figure 39. Altitude effects addition to the control output via a mechanical switch mechanism.

This problem is summarized on below paragraphs for better clarification. Notice the above figure 39 creates a function in the form of below equation 75.

$$O_f = O_1 - \frac{k}{HAG} \quad \text{(Equation 75)}$$

In the above equation, the terms are:

- O_f : Final control decision after the switch,
- O_1 : Original control decision by the decision field,
- k : A constant parameter to adjust model output,
- HAG : Altitude in ft, in form of height above ground.

Above logic simply subtracts an amount of control activity inversely proportional from O_1 to obtain O_f when the control output is positive and sums them up when the control output is negative. Best location for such a switch block is the motor blocks in the architectures so that the negative and positive channels do not need to be modified separately. Therefore the yellow box symbolising the effects of emotion and training are connected to the motor output channels in section 4.4 where step 3 model architecture is given.

The above-mentioned method functions well in an acceptable manner. The next paragraph will provide a brief discussion on the mathematical comparison between the DFT model and the human pilots given previously in section 5.1.3. But one important fact

is that above model in fact does not cover the original cortical behaviour. Please refer to below figure to observe how a field should in fact be modified to be biologically based.

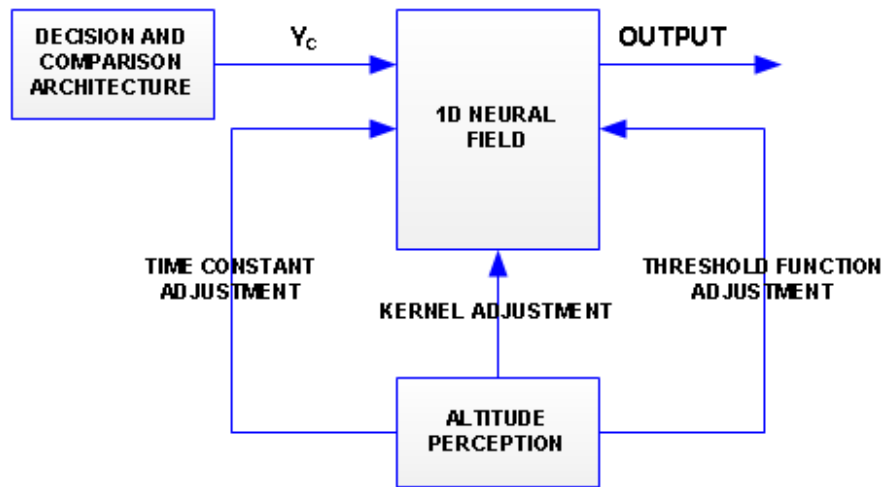


Figure 40. Biologically based adaptive field approach.

As it can be seen in figure 40, a biologically more plausible approach shall include adaptive parameter adjustment capability for neural fields so that the emotional effects can be controlled and modelled. While CEDAR does not allow such an application, DFT approach is not limited with that software and in MATLAB/SIMULINK or with any programming language one can write adaptive neural field models. This can be a subject for next phases of this study.

5.2.3 Evaluation of DFT Approach for Pilot Modelling

At the beginning of the 5th chapter, a correlation between the model 3's and the human pilots' performances was given as acceptable ($r^2 \approx 0.34$). Therefore one can argue that a DFT based pilot model is usable and possible despite the weaknesses of the framework summarized in sections 5.2.1 and 5.2.2. But, still there are some points to discuss about the possibility of a far more human-like pilot model using DFT. In the following paragraphs, some important points and criticisms about the work done in this study are summarized in the form of questions and answers. In other words, we attempt to propose some answers to those questions that arose from our model building and analysis efforts described in the previous chapters.

Can the neural field parameters of the model be further improved? Can a better performance be possible for the model step 3?

Basically, a better performance can be obtained, but if the question was about how easy would it be obtain that improvement in performance, it would be difficult to provide a positive answer. The main effector of the performance adjustment is the complexity of the model. First of all, current DFT frameworks do not allow the architect to implement an adaptive parameter shift in neural fields' exhibititive or inhibitive parameters. Due to this gap, the models shall be constructed in a feed forward structure and parameters shall be adjusted manually. In the step 3 model given above, 25 different 1D neural fields, 6 different rate-to-space or space-to-rate code transformations, 6 random Gaussian shapes,

14 network units such as summers and spatial dimensions resizing, and finally 3 different memory fields including individual pre-shapes are used. Omitting the summers and the Gaussians (which are in some cases used to create random atmospheric effects such as wind) since they are either non-parametric or user selected parametric units relatively easier to use, the total number of fields and field relevant architectural units becomes 41. Due to the number of units used in the model, important field parameters such as time constants, exhibitive and inhibitive kernel strengths, resting levels and threshold function parameters are above 250, excluding network connections and organisation. Therefore, in fact, after the construction of the architecture, optimization of the model parameters arises as an important challenge.

There would be various methods of optimising such a model to reach or simulate human performance. One would be -if it was allowed by and implemented in the framework- using a “parameters adjustment menu” which will allow adjustment of parameters manually while observing the effects on other connected fields and network elements in the same graphical user interface. The selected framework of this study did not allow the observation and manipulation of multiple model parameters in the same screen. Beyond this fact, the common limitation of all existing DFT frameworks on the adaptive parameter shifting or changing incapacitates all possible architectures on the aspect of “feedback learning”. i.e. the network itself is unable to learn from its experiences. Even if the human agent network does learn or not that way, for allowing networks or models with bigger sizes and dimensions, neural field models containing interfaces for adaptively modifiable field parameters are required. This learning mechanism does not need to be used for the implemented model’s experimental evaluation phase, but it can be a part of the architectural implementation process for selecting perfect parameters. Please see above figure for a suggested adaptive field approach.

Returning to our question, it is possible to make an adjustment for a better human-like pilot. But still the model needs to be further improved to go closer to human cognition since there is various spatiotemporal control tasks omitted here. Also some subtasks which are subparts of the main tasks (such as head turn control or eye gaze shift) are all neglected in the implementation presented in this study. At the end, a symbolic architecture would also omit such steps probably. But when the claim is to implement an “embodied” model, neglecting of some biological aspects would make the value of the model questionable.

Can there be a robotic implementation of the model? Would the DFT approach work for an autopilot? Is there a difference between implementing a DFT autopilot and trying to reach human performance?

To be able to implement a DFT based model embedded in a robotic application (i.e. online, on-board and autonomous) the model needs to be implemented in a language that allows building inside a programmable unit such as a microprocessor, a FPGA or a microcontroller. Currently most popular applications use C/C++ (or derivatives of C such as microC or picC for relatively simpler hardware), although in the past Assembly was popular for civilian applications, whereas ADA or Jovial were used for military applications. In the last decade some examples implemented in “Java for Embedded” can also be observed. One popular method is to implement the model in MATLAB/Simulink and convert it into C using an automatic code generator embedded inside the software. Clearly today the software development environments and embedded hardware

technology allows the developer to implement software with intensive cognitive capabilities. So what should be questioned is the maturity of the DFT approach and its capability on developing human-like agent functions.

The architecture and models given in this document can be used on the development of a DFT based autopilot. But it shall be underlined that, the conventions and regulations on the aviation area push very high safety considerations on autonomous functions and even for human-in-the-loop controls. The suitability of a feed forward architecture with no feedback control to ensure the stability of the system such as the one given in this study may be unsuccessful in compliance with such conventions and regulations. Of course, since DFT approach uses the stable or steady states of a dynamical system as primary states of the system, it is not expected by the very nature of the DFT that the system runs into an unexpected behaviour. But at the end, the software and hardware implementation scheme of the planned system may run into crashes while the embedded DFT mathematics may not be the cause of the crash. To be able to use such a model in an autopilot to be used in a commuter, general aviation, transport or airliner category aircraft, i.e. an aircraft carrying a human inside, it shall be clearly proven that all states of the system including the algorithms, the software implementation and hardware are all clearly deterministic. The model would be used in a small size unmanned aerial vehicle with fine adjustment of parameters, but when we come to a larger and relatively high safety requirement task, a feedback-wise adaptive parameter adjustment mechanisms may become mandatory for success and shall be implemented in C/C++.

Under circumstances where the adaptive parameter learning capability issue is solved with an indigenous solution of DFT, still the model above may be far more complicated for very fine parameters adjustment due to very high number of neural fields. All previous robotics applications given in section 3.3.5 uses relatively lower numbers of fields since their goal is not to exactly match human cognitive spatiotemporal task performance (Bicho, Mallet and Schöner, 2000; Monteiro and Bicho, 2011; Bicho, Louro and Erlhagen, 2010; Schöner, Dose and Engels, 1995; Sandamirskaya, 2014). Regarding the requirements of a robotic autopilot application, there is no need to run the system in a human-like method and using DFT in a relatively smaller size for just performing related tasks may be very useful. Unifying the above adaptive method with a relatively smaller architecture and an implementation allocating negative spatial axis definition may end in a successful cognitive autopilot for the cases where the goal is not to implement completely human-like behaviour. But for the cases where the aim is to understand the human behaviour using experimental setups for comparing human performance with detailed cognitive models, such simplification may be out of consideration since human performance is far beyond the capabilities of such models.

As summarized in previous chapters, the symbolic architectures are critiqued for being disembodied and generally neglecting the biological layer. Also we have referred to some studies pointing out that some very famous cognitive architectures have been biologically addressing the wrong brain parts (Eliasmith, 2013). In contrast, DFT is embodied, but in comparison with the symbolic architectures, it is yet in an early stage due to the lack of mature frameworks for supporting large-scale implementation. Although CEDAR is a promising framework supplying various tools, the adaptive field modification capability is not implemented yet and a future integration plan of such capability is not publicly known.

Does the improved model converge towards human cognition? Does generating performance similar to the human agent provide sufficient grounds for the claim that a cognitive model resembles human cognition?

This is probably the most important question of this study. The DFT approach and the present tools in the market are very promising for developing cognitive models of human behaviour. Especially by neglecting the high burden and complexity of neural models that are formed of networks of single neurons (i.e. modelling the highway instead of modelling each car as stated in previous paragraphs), the DFT approach eases the cognitive robotics development considerably. But the model given in this study is much more dominated by the structure of the software framework due to the two major limitations given in sections 5.2.1 and 5.2.2.

Since current DFT applications require only positive axis description, the model of axis definition in the DFT pilot model needed to be divided into two parts, one being negative and one being positive. Of course, today there is no widely accepted theory regarding how the human brain processes such spatiotemporal axes, and it may well be the case that human cognition may rely on a single axis definition containing both directions from the origin. So even if the model step 3 would show a perfect match to human performance, this would not necessarily make it a true demonstration of human cognition, but would present a case that can generate exact human performance based on a method inspired by human cognition. Notice this is also an important philosophical question applicable to all cognitive modelling and neuroimaging studies.

Does “being embodied” mean “being biologically true” for the DFT pilot model developed in this study?

This question is highly relevant to the previous question. DFT is a good representative of the embodiment approach to cognition as it employs the Amari equation based model of cortical electrical behaviour to dynamically generate the symbolic spatial dimensions as stable states of interacting dynamical systems, rather than presupposing a specific internal symbolic structure. But regarding the blocks of all DFT based models given in this study, it is not possible to map all of the blocks into specific brain regions due to the reasons discussed in the following paragraphs.

First of all, neurobiological underpinnings of the human navigational properties and capabilities in the cortex have not been fully decoded yet, and this study does not aim to make any strong claims regarding the possible mappings between the blocks implemented in the DFT model and specific brain regions.. Instead, the focus was to use the DFT approach to generate a model of embodied flight control without using any symbolic representations. Despite the various limitations of the DFT approach and the moderate level of correlation between the model and the human pilot data, one can argue that the main goal of this study has been reached.. However, the biological addressing of such a complex task is far beyond the scope of this study and need insights from other disciplines and methods such as neuroimaging and neurobiology. In short, a model’s being cognitively capable does not mean being biologically true.

Secondly, by being embodied and neglecting the symbolic structures, the DFT approach denies all symbolic (and sometimes metaphysics) approaches assuming a mind concept extending out of the brain, which also results in a denial of algorithmic layers. In other

words, no if-else or case-switch mechanisms are allowed in DFT based models. Of course, accepting the principles of embodiment as captured by DFT forces the modeller to employ such an approach. But still due to the denial of such methods and the approach being in early periods of development, various functions that are in fact very easy to implement in symbolic architectures become very challenging for DFT models. Therefore, the model given in this study underestimated many parts of the pilot navigational tasks, which should be included in future work within a more mature DFT framework.

Finally, the models developed in this study contained a segregation between azimuth and vertical axes of spatial environment in all steps of perception, cognition and motor implementations. In other words, completely independent channels of cognitive processing are assigned to pitch and yaw channels. Additionally roll channels are neglected to simplify the construction of the modelling architecture. But it is well known, even for car drivers, when the pilot focuses on any distracting task, the main control function may deviate from the goal. Although this idea is not to be proven or scientifically supported here, it may be useful to assume that navigational tasks run in a unity in a human agent and a segregation into independent channels may not reflect the biological truth. Under such assumptions, it may be deduced that a model has a probability of being biologically incomplete even for the maturities it may have such as being completely embodied and matching human performance.

Do “3D” and “2D” space robotic applications cause a difference for DFT implementations?

Most of the given robotics applications in section 3.3.5 are performing their navigation tasks in 2D space and provide quite good performance. The current study contains a 3D navigation task forcing the modeller to implement parallel channels to divide the space into relatively easy to control axes. In other words, the 3D space is diminished into 2D spaces to be able to achieve control behaviour in a relatively easier way.

In the discussion of the previous question it has been stated that the channel segregation architecture may be the main reason behind this model being biologically incomplete. If the task were to be performed in 2D space, this segregation would be unnecessary. Therefore, it may be argued that increasing the task difficulty (such as increasing the space dimensions, field numbers, increasing tasks' number of categories etc.) may lead to difficulties for architectures to be truly biologically based.

5.2.4 Conclusion

In this thesis study a dynamic field theory (DFT) based cognitive model of a pilot is proposed by extending the existing DFT model building frameworks originally devised for developing 2D robotics applications. The implications of this work for cognitive science in general can be discussed in reference to an influential taxonomy of cognitive theories proposed by J.A. Fodor and Z.W. Pylyshyn (1988) In their influential paper Fodor and Pylyshyn (1988) divides all cognitive theories under two main groups namely one being the Representationalist and the other Eliminativist.

There are two major traditions in modern theorizing about the mind, one that we'll call 'Representationalist' and one that we'll call 'Eliminativist'. Representationalists hold that postulating

representational (or 'intentional' or 'semantic') states is essential to a theory of cognition; according to Representationalists, there are states of the mind which function to encode states of the world. Eliminativists, by contrast, think that psychological theories can dispense with such semantic notions as representation. According to Eliminativists the appropriate vocabulary for psychological theorizing is neurological or, perhaps behavioural, or perhaps syntactic; in any event, not a vocabulary that characterizes mental states in terms of what they represent (Fodor and Pylyshyn, 1988, p.7).

Regarding the above classification and the rest of the chapter, Fodor and Pylyshyn classifies the connectionist approaches under the representationalist group and argues that although connectionism denies the existence of semantic layers, it should still be considered under the representationalist camp due the coding of semantic knowledge hidden inside the syntax of the network. Even the sub-symbolic claims of connectionism are denied by Fodor and Pylyshyn (1988) on the grounds that such sub-symbolic approaches are often found acceptable by many Representationalists. This claim may also be appreciated by Schöner (2007) who claims that DFT method uses the neural activity as the representations of external world in modelling cognition. At the end, here in the study given in this thesis a connectionist model is provided, but the implementation does not include any algorithmic layer or a classical neural network, despite a dynamical field theory approach is used to construct a pilot model performing basic attitude control tasks. Therefore it may be deduced that this study provides a cognitive pilot model which may be categorised under the connectionist-representationalist architectures, but not necessarily under algorithmic cognitive architectures such as ACT-R. So it is also not a properly representationalist model. But yet it may be classified under hybrid models regarding the classifications of Wray and Jones (2000).

In the previous paragraphs the limitations of the architecture developed in this study were discussed. The DFT based pilot model developed in this study is able to control at least a simulated simple 3 DoF aircraft model in 2 axes one being pitch and one being yaw. In short, the final model, namely model step 3, is a DFT based and embodied pilot cognitive model simulating human vertical hold performance at a moderate level of correlation (~ 0.34) as compared to a human pilot, without specifically addressing the model's components to specific neurobiological components.

At the end, the model developed in this study suggests that the DFT approach can be employed to develop embodied cognitive robotics applications for navigating in 3D space by extending previous robotics applications generally implemented in 2D. Our findings point out the issues involved with such a transition and some strategies for dealing with them. By this property, the study tries to use an unexperienced methodology for a task older than 50 years using the knowledge supported by the previous 2D robotics applications.

Additionally it has been observed that with relatively simpler architectures, parameter adjustment may be simpler and such structures may lead robotics applications to succeed in more complicated tasks. Regarding the experience gained during this study, it may be argued that DFT is much more promising in the area of cognitive robotics in comparison with the experimental cognitive research area which uses cognitive modelling.

Although existing DFT modelling frameworks supported the study in significant ways, they also brought their own limitations in to the study. It has been observed that the DFT architectures, frameworks or all types of infrastructures using that approach may benefit in various aspects from a novel capability of neural fields with adaptively controllable parameters. Such an update may allocate the frameworks to enable adaptively learning architectures and ease the implementation of much more complicated tasks.

Finally it has been observed that even for completely embodied and very successful models, the DFT approach does not claim any biological addressing. Additionally general acceptance of embodiment concepts prevents the usage of symbolic-algorithmic functions. This commitment makes it difficult to implement complex functions such as a simple comparison algorithm between two single digit numbers, which may become a task of multiple neural fields in the DFT context. Under these circumstances, to close these functional gaps, DFT frameworks may be supported with further biological addressing to bring in some additional functional capability to the infrastructure. In other words, if a framework allows the user to use embedded functions of some expert biological parts in human brain, i.e. some cortical functions are by default provided to the user, these functional difficulties may be at least partially eliminated. To achieve such a goal, DFT researchers may focus on the implementation of cortical functions rather than robotic applications, but this goal does not seem to be reachable in the near future due to the very promising advantages of DFT in the robotics domain.

5.2.5 Future Work

The study given here is assumed to be a first and early part of a larger study. In the next stages, the DFT based pilot model will be improved with adaptive parameters control capability. Since the CEDAR framework does not support such a method, the next model is planned to be implemented in C language as a stand-alone application without making a commitment any existing DFT framework. In other words, a DFT solver will be written in C and the model will be developed in compliance with such a solver.

After the C implementation stage is complete, the model will be embedded into a small size commercial UAV. For the very first phases of UAV trials, the goals will be basic navigation tasks, obstacle avoidance and manoeuvre control. After achieving these goals, two UAV formations and dogfights will be investigated in the next stage. Although the DFT approach's denial of symbolic-algorithmic functions results in some computational limitations, its enormous performance on spatiotemporal tasks may allow such approach to be very useful for implementing complex 3D flying characteristics such as fighter dogfight.

In the final stage of the study, it is aimed to perform multiple UAV autonomous functions including; acquisition of hostile UAVs, tracking and attacking, defeating the attacks by manoeuvres and landing.

REFERENCES

Robbins P., Aydede M. (2009). *The Cambridge handbook of situated cognition*. Cambridge University Press. NY.

Hurzook A., Trujillo O., Eliasmith C. (2013). Visual motion processing and perceptual decision making. *Proceedings of 35th Annual Conference of the Cognitive Science Society*. pp.2590.

Schöner, G. (2007) Development as Change of System Dynamics: Stability, Instability, and Emergence. In J.P. Spencer, M. Thomas, & J. McClelland (Eds.), *Toward a New Grand Theory of Development? Connectionism and Dynamic Systems Theory Re-Considered*, Oxford University Press.

Amari S.I. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, vol. 27, p.77–87.

Fodor J.A., Pylyshyn Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. Retrieved June 18, 2015 from online archive website <http://www.sciantaanalytics.com/sites/default/files/fodor-pylyshyn.pdf>.

Blakelock J.H. (1991). *Automatic Control of Aircrafts and Missiles* (2nd ed.), John Wiley and Sons publishing, New York.

McRuer D.T. And Jex H.R. (1967). A Review of quasi-linear pilot models. *IEEE Transactions in Human Factors in Electronics*, vol. 8. pp.231-249.

Anderson R.O. (1970). A new approach to the specification and evaluation of flying qualities. AFFDL-TR-69-120. USA Air Force Flight Dynamics Laboratory Report.

Anderson R.O., Connors A.J. and Dillow J.D. (1970). Paper pilot ponders pitch. AFFDL-TR-69-120. USA Air Force Flight Dynamics Laboratory Report.

Pollard J.J. (1975). All digital simulation for manned flight in turbulence. AFFDL-TR-75-82. USA Air Force Flight Dynamics Laboratory Report.

Johnson N.E., Pritchett A.R. (2002). Generic Pilot and Flight Control Model for use in simulation studies. In AIAA Modelling and Simulation Technologies Conference and Exhibit. USA.

Hutchins E. (1995). How a cockpit remembers its speeds. *Cognitive Science Vol 19*. pp.265-288.

Clark A. (2008). *Supersizing the Mind, Embodiment, Action and Cognitive Extension*, Oxford University Press, New York USA.

Hutchins E. (2010). Enculturating the Supersized Mind. In *Philosophical Studies, vol. 152*. pp.437-446. Retrieved September, 01, 2015 from Springerlink.com.

Eliasmith C. (2013). *How to build a brain. A Neural architecture for biological cognition*. Oxford University Press.

Hutchins E. (2000). The Cognitive consequences of patterns of information flow. *Intellectica Vol 1:30*. pp. 53-74.

Hutchins E., Klausen T. (1996). Distributed cognition in an airline cockpit. In Y. Engeström and D. Middleton (Eds.), *Cognition and communication at work*. New York: Cambridge University Press. pp. 15-34.

Wiegmann D.A., Shappell S.A. (2000). Human error perspectives in aviation. *The International journal of aviation psychology. Vol 11(4)*. pp.341-357.

Ye N. (2003). An Information processing model for human-computer integrated assembly planning systems. *Journal of the Chinese institute of industrial engineers. Vol 20, no 3*. pp.220-229.

Hutchins E., Holder B.E., Pérez A.R. (2002). Culture and flight deck operations. Boeing Company report.

Nomura S., Hutchins E. (2006). The multimodal production of common understanding in intercultural flight deck training. Boeing Company report.

Nomura S., Hutchins E., Holder B.E. (2006). A multi-cultural study of paper use in the flight deck. Boeing Company report.

Hutchins E. (2001). How pilots represent risk. In *CNRS Workshop on Risk*, Gif-surYvette, France.

Hutchins E., Holder B. (2000). Conceptual models for understanding an encounter with a mountain wave. *Proceedings of HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics*, Toulouse, France.

Holder, B and Hutchins, E. (2001). What Pilots Learn About Autoflight While Flying on the Line. *Proceedings of the 11th International Symposium on Aviation Psychology*, Columbus Ohio.

Hutchins, E., Holder, B., and Hayward, M. (1999). Pilot attitudes toward automation. Retrieved August 10, 2015 from UCSD website for Hutchins E.'s publications, <http://hci.ucsd.edu/hutchins/attitudes/index.html>.

Taatgen N., Anderson J.R. (2009). The Past, present and the future of cognitive architectures. in *Topics in Cognitive Science. Vol 2009. pp.1-12.*

Marr D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. New York: Freeman.

Turing A.M. (1950). Computing machinery and intelligence. in *Mind, vol 59. pp.433–460.*

Newell, A., & Simon, H. (1963). GPS, a program that simulates human thought. In *E. A. Feigenbaum & J. Feldman (Eds.), Computers and thought* (pp. 279–293). New York: McGraw- Hill.

Laird J. E., Newell A., Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, vol 33. pp.1–64.*

Wray R.E., Jones R.M. (2006). Considering SOAR as an agent architecture. In Sun R. (Ed.), *Cognition and multiagent interaction: From cognitive modelling to social simulation*. Cambridge University Press.

Anderson J.R., Bothell D., Byrne M.D., Douglass S., Lebiere C. And Qin Y. (2004). *Psychological Review*. Vol 111. No:4. pp.103-1060.

Newell, A. (1990). Unified theories of cognition. Cambridge, MA: Harvard University Press.

Anderson J.R. (2002). ACT: A Simple theory of complex cognition. In Polk T.A. and Seifert C.M. (Eds.), *Cognitive Modelling*. The MIT Press.

Langley P., Laird J.E., Rogers S. (2008). Cognitive architectures: Research issues and challenges. In *Cognitive Systems Research (2008)*, doi:10.1016/j.cogsys.2006.07.004.

Taatgen N., Lebiere C., Anderson J.R. (2006). Modelling paradigms in ACT-R. In Sun R. (Ed.), *Cognition and multiagent interaction: From cognitive modelling to social simulation*. Cambridge University Press.

Rosenbloom P.S., Laird J.E., Newell A., McCarl R. (2002). A Preliminary Analysis of SOAR architecture as a basis for general intelligence. In Polk T.A. and Seifert C.M. (Eds.), *Cognitive Modelling*. The MIT Press.

Spencer R.M.C., Karmarkar U., Ivry R.B. (2009). Evaluating dedicated and intrinsic models of temporal encoding by varying context. In *Philosophical Transactions of The Royal Society B*. vol (2009) 364. pp.1853-1863.

Enns R., Si J. (2004). Helicopter Flight Control Using Direct Neural Programming. In Si J., Barto A., Powell W. and Wunsch D. (Eds.), *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press and John Wiley and Sons Inc.

Kaneshige J., Bull J., Totah J.J. (2000). Generic Neural flight Control and Autopilot System. AIAA-2000-4281.

Kaneshige J., Burken J. (2008). Enhancements to a Neural Adaptive Flight Control System for a Modified F-15 Aircraft. In *Proceedings of the 2008 AIAA GNC Conference*.

Venkov N. A. (2008). Dynamics of Neural Field Models. University of Nottingham Doctoral Thesis Retrieved May 27, 2015 from Researchgate archive http://www.researchgate.net/publication/265224113_Dynamics_of_Neural_Field_Models.

Bear M. F., Connors B. W., Paradiso M. A. (2001). Neuroscience: exploring the brain. Lippincott Williams and Wilkins.

Hodgkin A. L., Huxley A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. In *Journal of Physiology*, vol 117. pp.500–544.

Morris C., Lecar H. (1981). Voltage oscillations in the barnacle giant muscle fiber. In *Biophysical Journal*, vol. 35. pp.193–213.

Gerstner W., Kistler W. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, 2002.

Jolivet R., Lewis T. J. Gerstner W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. In *Journal of Neurophysiology*, vol. 92. pp.959–976, 2004.

Bressloff P. C., Coombes S. (2000). Dynamics of strongly coupled spiking neurons. In *Neural Computation*, vol. 19. pp.91–129.

Brette R., Gerstner W. (2005) . Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. In *Journal of Neurophysiology*, vol.94. pp.3637–3642.

Eliasmith C., Anderson C. H. (2003). Neural engineering. Computation, representation and dynamics in neurobiological systems. Bradford Books, MIT Press.

Eliasmith C., Stewart T. C., Choo X., Bekolay T., DeWolf T., Tang Y., Rasmussen D. (2012). A Large-scale model of the functioning brain. In *Science* vol. 338. pp. 1202-1205.

Choo X., Eliasmith C. (2014). General instruction following in a large-scale biologically plausible brain model. Retrieved July 19, 2015 from Cognitive Science

Society's journal TOPICS website, online archive
<http://csjarchive.cogsci.rpi.edu/Proceedings/2013/papers/0084/paper0084.pdf>.

Bekolay T., Bergstra J., Hunsberger E., DeWolf T., Stewart T. C., Rasmussen D., Choo X., Voelker A.R., Eliasmith C. (2014). Nengo.: A Python tool for building large-scale functional brain models. In *Frontiers in Neuroinformatics. Vol. 7*. Article 48. pp.1.13.

Stewart T. C., Eliasmith C. (2011). Neural cognitive modelling: a biologically constrained spiking neuron model of Hanoi tower task. In *Proceedings of CogSci 2011*. Retrieved July 22, 2015 from Academia.edu online archive http://www.academia.edu/2657834/Neural_cognitive_modelling_a_biologically_constrained_spiking_neuron_model_of_the_Tower_of_Hanoi_task.

Aaronson S. (2011). Why philosophers should care about computational complexity. <http://arxiv.org.1108.1791v3>.

Cuijpers. R.H., Erhagen W. (2008). Implementing Bayes rule with neural fields. In Kurkova et al. (Eds), *ICANN 2008, Part II, LCNS 5164*. pp. 228-237.

Schöner G. (2008). Dynamical systems approach to cognition. In Sun R. (Ed.) *The Cambridge handbook of computational psychology*. Cambridge University Press.

Anderson M. L., Richardson M. J., Chemero A. (2012). Eroding the boundaries of cognition: Implications of embodiment. In *Topics in Cognitive Science, vol 201*. pp.1-14.

Lomp O., Zibner S. K. U., Richter M., Schöner G. (2013). A software framework for cognition, embodiment, dynamics and autonomy in robotics: Cedar. In Mladenov V., Koprinkova-Hristova P., Palm G., Villa A.E.P, Apollini B., and Kasabov N. (eds), *Artificial Neural Networks and Machine Learning, ICANN 2013*, p. 475-482. Springer Berlin Heidelberg.

Coombes S. (2012). Neural fields. Retrieved August 20, 2015 from Scholarpedia website archive. <http://www.scholarpedia.org>.

Beurle R. L. (1956). Properties of a mass of cells capable of regenerating pulses. In *Philosophical Transactions of Royal Society of London B, vol. 240*. pp.55–94.

Wilson H.R. and Cowan J.D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. In *Biophysics Journal*, vol. 12. pp.1–24.

Bicho E., Louro L., Erlhagen W. (2010). Integrating verbal and non-verbal communication in a dynamic neural field for human-robot interaction. In *Frontiers in Neurorobotics*, vol. 4 Article 5. pp.1-13.

Pinotsis D.A., Moran R.J., Friston K.J. (2012). Dynamic causal modelling with neural fields. In *Neuroimage*, vol. 59. pp.1261-1274.

Potthast R., Graben P.B. (2008). Existence and properties of solutions for neural field equations. In *Mathematical Methods in the Applied Sciences*. Retrieved August, 21, 2015 from www.interscience.wiley.com.

Satel J., Story R., Hilchey M.D., Wang Z., Klein R.M. (2013). Using a dynamic neural field model to explore a direct collicular inhibition account of inhibition of return. Retrieved August 18, 2015 from <http://arxiv.org>. 1307.5684v1.

Van Gelder T. (1998). The dynamical hypothesis in cognitive science. In *Behavioral and Brain Sciences*, vol 21. pp. 616.665.

Bicho E. (2000). *Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Shaker Verlag, Aachen.

Palmer S.E. (1999). *Vision Science: From Photons to Phenomenology*. MIT Press: Cambridge, MA.

Erlhagen W. (2002). Internal models for visual perception. In *Biological Cybernetics*, vol 88. pp. 409-417.

Johnson J.S., Spencer J.P., Schöner G. (2007). Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. In *New Ideas in Psychology*, Retrieved July 12, 2015 from website of Journal. doi:10.1016/j.newideapsych.2007.07.007.

Jancke D., Erlhagen W., Dinse H.R., Akhavan A.C., Giese M., Steinhage A., Schöner G. (1999). Parametric population representation of retinal location: neuronal

interaction dynamics in cat primary visual cortex. In *The Journal of Neuroscience*, vol 19(20). pp.9016-9028.

Simmering V.R., Spencer J.P. (2008). Generality with specificity: the dynamic field theory generalizes across tasks and time scales. In *Developmental Science* vol.11:4. pp. 541–555.

Simmering V.R., Schutte A.R., Spencer J.P. (2007). Generalizing the dynamic field theory of spatial cognition across real and developmental time scales. In *Brain Research* vol. 2007, doi:10.1016/j.brainres.2007.06.081.

Spencer J.P., Simmering V.R., Schutte A.R., Schöner G. (2007). What does theoretical neuroscience have to offer the study of behavioural development. In *The emerging spatial mind*. Oxford University Press. pp.320-261.

Sandamirskaya Y., Schöner G. (2006). Dynamic field theory and embodied communication. In I. Wachsmuth, & G. Knoblich (Eds.), *Lecture notes in artificial intelligence: Vol. 4930. Modeling communication with robots and virtual humans*. pp. 260–278. Springer.

Spencer, J.P., Schöner, G. (2003) Bridging the representational gap in the dynamical systems approach to development. In *Developmental Science* vol. 6. pp. 392–412.

Erlhagen W., Jancke D. (2004). The role of action plans and other cognitive factors in motion extrapolation: a modelling study. In *Visual Cognition*, vol 11(2/3). pp.315-340.

Bastian A., Schöner G., Riehle A. (2003). Preshaping and continuous evolution of motor cortical representations during movement preparation. In *European Journal of Neuroscience*, vol. 18. pp. 2047-2058.

Erlhagen W., Schöner G. (2002). Dynamic field theory of movement preparation. In *Psychological Review*, vol. 109. No:3. pp.545-572.

Schutte A.R., Spencer J.P., Schöner G. (2003). Testing the dynamic field theory: working memory for locations becomes more spatially precise over development. In *Child Development*, vol. 74, no:5. pp.1393-1417.

Sandamirskaya, Y., Schöner, G. (2010). An embodied account of serial order: How instabilities drive sequence generation. In *Neural Networks*, vol. 23(10). pp. 1164–1179.

Sandamirskaya, Y., Schöner, G. (2010b). Serial order in an acting system: a multidimensional dynamic neural fields implementation. In *Development and Learning, ICDL 2010*. 9th IEEE International Conference on.

Duran B., Sandamirskaya Y., Schöner G. (2011). A dynamic field architecture for the generation of hierarchically organized sequences. In *International Conference on Artificial Neural Networks*.

Sandamirskaya Y., Storck T. (2014). Learning to look and looking to remember: a neural-dynamic embodied model for generation of saccadic gaze shifts and memory formation. In *Springer Series in Bio-/Neuroinformatics*, vol.4. pp. 175-200.

Kazerounian S., Luciw M., Richter M., Sandamirskaya Y. (2013). Autonomous reinforcement of behavioural sequences in neural dynamics. Retrieved June 02, 2015 from <http://arxiv.org.1210.3569v2>.

Sandamirskaya Y. (2013) Dynamic Neural Fields as a Step Towards Cognitive Neuromorphic Architectures. In *Frontiers in Neuroscience*, vol. 7:276. doi:10.3389/fnins.2013.00276.

Sandamirskaya Y., Schöner G. (2010c). Dynamic Field Theory of Sequential Action: A Model and its Implementation on an Embodied Agent. In *7th International Conference on Development and Learning*.

Richter M., Sandamirskaya Y., and Schöner G. (2012). A robotic architecture for action selection and behavioral organization inspired by human cognition. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE*. pp. 2457–2464.

Sandamirskaya Y., Richter M., Schöner G. (2011). A neural-dynamic architecture for behavioral organization of an embodied agent. In *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL EPIROB 2011)*.

Sandamirskaya Y. (2011). Learning in Dynamic Neural Fields Model for Sequence Generation. In *DevLeaNN 2011: A Workshop on Development and Learning in Artificial Neural Networks*. Retrieved August 29, 2015 from <http://devleann.iscpif.fr>.

Duran B., Sandamirskaya Y. (2011). Neural Dynamics of Hierarchically Organized Sequences: a Robotic Implementation. Retrieved August 03, 2015 from online archive http://www.academia.edu/11403851/Neural_dynamics_of_hierarchically_organized_sequences_a_robotic_implementation.

Schutte A.R., Spencer J.P. (2007). Planning “discrete” movements using a continuous system: insights from a dynamic field theory of movement preparation. In *Motor Control*, vol. 11. pp.166-208.

Batiha B., Noorani M.S.M., Hashim I. (2008). Numerical Solutions Of The Nonlinear Integro-Differential Equations. In *International Journal of Open Problems Comp. Math.*, Vol. 1, No. 1.

Soliman A.F., EL-ASYED A.M.A., El-Azab M.S. (2012). On The Numerical Solution of Partial integro-differential equations. In *Mathematical Sciences Letters*. Vol. 1, no: 1. pp.71-80.

Tang T. (1993). A finite difference scheme for partial integro-differential equations with a weakly singular kernel. In *Applied Numerical Mathematics*, vol. 11. pp.309-319.

Meijer H.G.E., Coombes S. (2013). Travelling waves in a neural field model with refractoriness. In *Journal of Mathematical Biology*, vol 68. pp. 1249-1268.

Coombes S. (2005). Waves, bumps, and patterns in neural field theories. In *Biological Cybernetics*, vol. 93. pp.91-108.

Erlhagen W., Bicho E. (2006). The dynamic neural field approach to cognitive robotics. In *Journal of Neural Engineering*, vol. 3, no: 3. pp. R36–R54.

Taatgen N.A., Anderson J., Van Rijn H. (2007). An integrated theory of prospective time interval estimation: the role of cognition, attention and learning. In *Psychological Review*, vol.114, no:3. pp. 577-598.

Taatgen N.A., Van Rijn H. (2011). Traces of times past: representations of temporal intervals in memory. In *Mem. Cogn.*, vol.39. pp.1546-1560.

Berlin H.A., Rolls E.T., Kischka U. (2004). Impulsivity, time perception, emotion and reinforcement sensitivity in patients with orbitofrontal cortex lesions. In *Brain*, vol.127. pp.1108-1126.

Karmarkar U.R. (2011). Defining the contributions of network clock models to millisecond timing. In *Frontiers in Integrative Neuroscience*, vol. 5, article 41. pp.1-2.

Karmarkar, U. R., Buonomano, D.V. (2007). Timing in the absence of clocks: encoding time in neural net-work states. In *Neuron*, vol.53. pp.427–438.

Medina J.F., Garcia K.S., Nores W.L., Taylor N.M., Mauk M.D. (2000). Timing mechanisms in the cerebellum: testing predictions of a large-scale com-puter simulation. In *Journal of Neuroscience*, vol.20. pp.5516–5525.

Leon M.I., Shadlen M.N. (2003). Representation of Time by Neurons in the Posterior Parietal Cortex of the Macaque. In *Neuron*, vol 38. pp.317-327.

Muratov S., Lakhman K., Burtsev M. (2014). Neuroevolution of sequential behavior in multi-goal navigation task. In *ALIFE 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*.

Bicho E., Mallet P.,Schöner G. (2000). Target representation on an autonomous vehicle with low-level sensors.In *The International Journal of Robotics Research*, vol. 19. pp. 424–447.

Hemion N.J. (2013). Building block for cognitive robots: Embodied simulation and schemata for cognitive architecture. Der Technischen Fakultät der Universität Bielefeld. Doctoral thesis retrieved August 19, 2015 from academics thesis archive of Bielefeld University. <http://pub.uni-bielefeld.de/publication/2643718>.

Monteiro S., Bicho E. (2009).Attractor dynamics approach to formation control: theory and application. In *Autonomous Robots*, vol. 29(3-4).pp.331-335. Springer.

Carretero J.G.H., Nieto F.J.S., Cordón R.R. (2012). Aircraft trajectory simulator using a three degrees of freedom aircraft point mass model. In *Proceedings of the 3rd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS'2013)*.

Schöner G., Dose M., Engels C. (1995). Dynamics of behavior : theory and applications for autonomous robotic architectures. In *Robotics and Autonomous Systems*, vol.16. pp.213-245.