Qualitative Modeling of Biological Neural Network: Implementation on Basal Ganglia

by

Ayesha Gohar

2010-NUST-MS-M&S-16

A Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTATIONAL SCIENCE AND ENGINEERING

Thesis Supervisor

Dr Jamil Ahmad

Research Centre for Modeling and Simulation

National University of Sciences & Technology

2012

بسم الله الرحمن الرحيم

In the name of Allah, the most

Beneficent and the most Merciful

DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Dr. Jamil Ahmad. All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Ayesha Gohar

ACKNOWLEDGEMENTS

Immeasurable words of praise and thanks to Allah, the Almighty, and the Creator of the universe for carving the path for me and always guiding and helping me out in the best possible way. Without His Will and Mercy, I would have never been able to accomplish this milestone. I am grateful to my beloved parents, Mr. & Mrs. Gohar Aman for their immense prayers, love, moral and undivided support, throughout my academic career and inspired me to pursue degree in Computational Science & Engineering (CS&E).

I am truly grateful to my supervisor, Dr. Jamil Ahmad, for his supervision, guidance, and patience starting from the initial advice and project plans, through ongoing advice, till the successful ending of this project. I am highly thankful to the principal, Dr. Ahmed Ejaz Nadeem for his continuous and valuable suggestions and guidance, especially for providing me the best environment and facilities. I would like to acknowledge the financial and academic support of RCMS (NUST) and its staff, particularly in awarding me Postgraduate scholarship.

I owe sincere and earnest thankfulness to my committee members Engr. Sikandar Hayat Mirza and Muhammad Tariq Saeed. Along with this, I Thank Dr. Umar Khan Niazi for his valuable suggestions and comments which were a great source of input to improve the research work presented in this thesis.

Special thanks to my beloved husband Amad Ullah babar for supporting me in the project and making it a success. I also acknowledge the support I have been receiving from my dear brothers, Arsalan & Usman, my sweet sister Marjan and my in-laws. Finally I thank my fellow friends: Nida, Shafia, Madiha & Sadia.

DEDICATION

To My Family for their Prayers without which I would not have been able to carry out my studies

ABSTRACT

Biological Neural Networks (BNN's) are complex systems represented as directed graphs where nodes represent neurons and edges represents interaction (activation or inhibition) between neurons. In order to analyze such system, we opt for hybrid modeling which incorporates the discrete as well as continuous behaviors of BNN. For discrete modeling, we use the well-known René Thomas' qualitative modeling approach which takes a set of logical parameters to predict the behavior of the Biological Regulatory Networks (BRNs). Basal Ganglia is a biological neural network consisting of Striatum (D1 and D2), Thalamus, Globus Pallidus internal (GPi), Globus Pallidus external (GPe), Sub Thalamic Nucleus (STN), Substantia Nigra and is strongly connected with cortex. In this thesis, we construct a qualitative model of the BNN by using GENOTECH tool in which we observe various behaviors in the form of cycles and stable states. Due to the total abstraction of time in qualitative modeling, we incorporate two types of delays in this model: activation delay (d_u) and inhibition delay (d_l). This results in a hybrid model of Basal Ganglia for which the state of the art, Hybrid model checking tool, HyTech is used to analyze the model. We incorporate the above approach on basal ganglia to characterize its behavior associated with thalamus and hence cortex; a key area in brain involved in actions related to motor neurons. For validation of our work, we identified different pathways and corresponding delay constraints which may lead to the development of tremors which is a disease state. Hytech synthesizes delay constraints characterizing different cyclic and diverging trajectories towards stable states. The stable state shows that the system converges towards tremors. As delays are used as parameters in our model, we conclude that if a set of values of parameters satisfies the constraints that follow a particular path, it will remain in that cycle else will follow another path or a deadlock state.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Biological Neural Networks

Brian tissue consists of around 100 billion [3] of nerve cells called neurons. Interconnection among these neurons makes brain a very complicated system to handle and analyze fully. The exact behavior of such a system is still not fully understood. Due to recent advance in neurobiology and rigorous growth in computational capabilities, neurobiologists have been able to develop more realistic and dynamic biological neuronal networks (BNN's) models. For a complete BNN model for analysis, it must have the known pathophysiology, neuroanatomy, network electrophysiology, pharmacology, and behavioral findings. Due to the vast inter connections of the brain, the formulation of the complete model is computationally restricted; however new insights have been proposed into brain function due to the scaled and abstract models of anatomical subsystems.

Many neurological disorders are essential cause of morality and non-communicable conditions among which cerebrovascular disease, Parkinson, Alzheimer and other dementia are on the top of the list. Such selected neurological disorders are responsible for an estimated 12% of total deaths. Within neurological disorders, 8.41% of it constitutes of Alzheimer and Parkinson diseases [1]. Basal ganglia is an attractive system to model and plays a central role in the development of many neurological disorders like Huntington and Alzheimer's disease. Therefore a lot of research is going regarding the pathology and working of basal ganglia, in order to address these diseases.

1.2 Motivation

Brain receiving, analyzing and transmitting information from one part of the body to another in such a short span of time, does not only show the complexity of the system but also one gets inquisitive about its nature and how it works. It's fascinating how it handles multiple processing simultaneously and avoids unwanted signals and if fault occurs in a single neuron, it disturb the entire system and ends up in severe diseases like Huntington's and Parkinson's disease. Basal ganglia is an important part of brain responsible for proper functioning of motor neurons due to which most of the body movements occur. Loss of such abilities may occur to improper functioning of basal ganglia mainly due to loss of neurons in a specific section of basal ganglia.

1.3 Background

1.3.1 Neuron

The brain consists of seven main lobes (Figure 1): Frontal Lobe, parental lobe, occipital lobe, temporal lobe, pones and cerebellum. Each lobe is made up of nerve cells called neuron. Neuron has three main parts (Figure 2):

a) Dendrites, one that receives information (receives action potential)

b) Cell body (soma), one that decides whether an action potential must be transmit or not

c) Axon, ending of neuron which transmits from cell body to the ends of neuron and from where it is transmitted to another neuron.

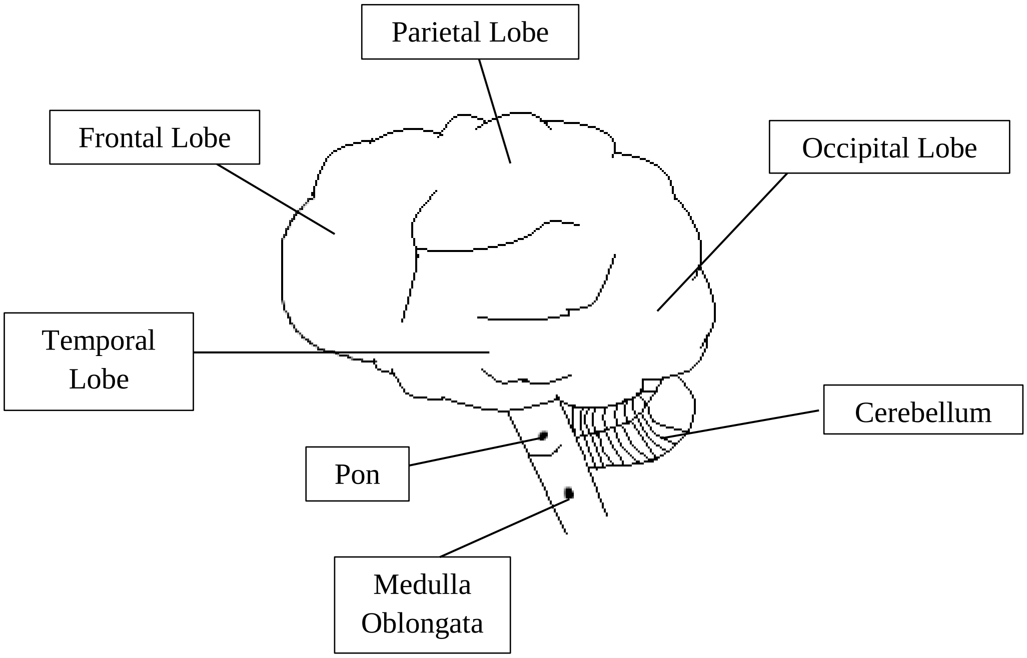


Figure 1: Major parts of brain: Frontal Lobe, Parietal Lobe, Occipital Lobe, Cerebellum, Medulla Oblongata, Pon, Temporal Lobe, and Frontal Lobe

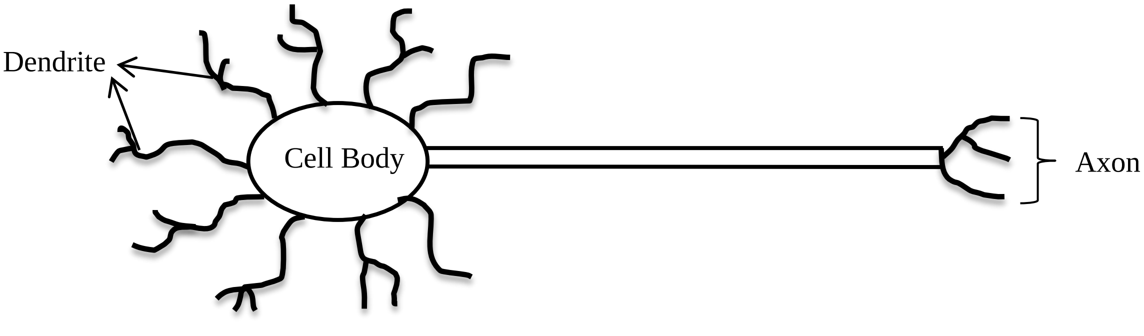


Figure 2: A single neuron representing three main parts: dendrites, cell body and axon

Neurons main function is to transmit information that is signal called action potential, to and fro brain. Action potential has been formalized in variety of methods for example symbolic analysis [2]. Our focus in this paper will be on it discrete modeling and how delay in transmission can affect the normal behavior of the system and end up in oscillations i.e. tremors.

Synapse is a small junction between an axon of one neuron (transmitting) and dendrite of another neuron (receiving). These are two fundamental types of neurons called chemical synapse and electrical synapse. In chemical synapse [Figure 3], a chemical is released by Presynaptic membrane called neurotransmitter. It binds to the receptors in postsynaptic membrane causing either an activation of electrical or a secondary messenger pathway.

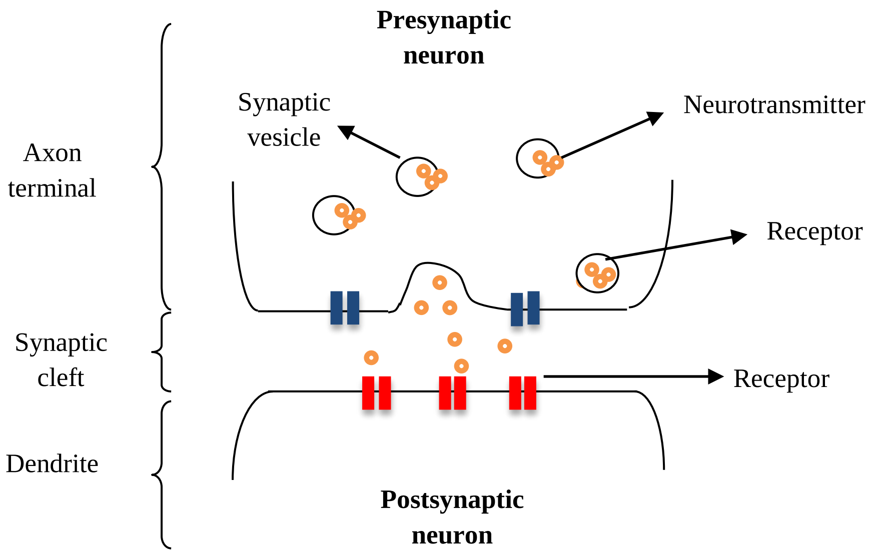


Figure 3: Chemical Synapse showing the interaction between pre and post synaptic membrane

1.3.2 Action Potential

Major function of neurons is transmission of information from one part of the brain to another in form of action Potential. Action potential is an event following a particular trajectory in which electrical potential across membrane rises and falls resulting in flow of information across the neuron.

In cell plasma, there are different types of voltage-gated ion channels embedded which are responsible for generation of action potential in cells. In resting state and stimulating phase, these channels are closed until it reaches the near the threshold potential of the cell. Once it crosses the exact membrane potential which is called threshold, they quickly begin to open. When the channels open, sodium ions flow inside the cell due to which an electrochemical gradient is produced resulting rise in the membrane potential. The electric current increases with time as more and more channel to open, resulting in rise of potential. Due to this process, there is a huge increase in membrane potential until all of the available ion channels are open. The rapid inflow of sodium ions causes the polarity of the plasma membrane to converse, and the ion channels then swiftly deactivate. When the sodium channels close, sodium ions can no longer enter the neuron, and they are keenly conveyed out of the plasma membrane. After this, potassium channels are then activated and they open, resulting in an outward current of potassium ions causing the electrochemical gradient to the resting state. After an action potential has transpired, there is the after hyperpolarization or refractory period which is transient negative shift. This occurs due to additional potassium currents that flow inside the cell. This is the mechanism that prevents an action potential from traveling back the way it just came.

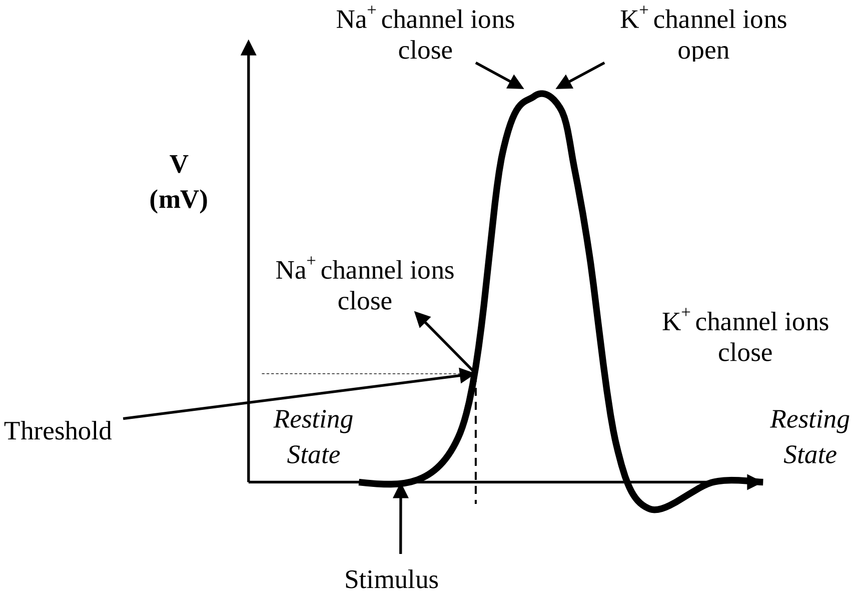


Figure 4: Action potential showing opening and closing of sodium (Na^+) and potassium channels (K^+)

1.3.3 Basal Ganglia

The anatomy and physiology of basal ganglia has been well described yet a very complex system with many hypothesis of its model. The basal ganglia consist of four main subnuclei[3, 4, 5]: striatum, globus pallidus [internal segment (GPi) and external segment (GPe)], subthalamic nucleus (STN), and substantia nigra [compact (SNc) and reticular (SNr)]. Major role of this system is the smoothing of voluntary movements. Movement disorders such as Parkinson’s disease (PD) and Huntington’s disease have been traced to basal ganglia dysfunction. Therefore, the basal ganglia are classified as part of the extrapyramidal motor system.

Parkinson’s disease is characterized by the loss of dopamine generating cells in substantia nigra (SNc). Early conceptual models (DeLong, 1990) diagrammed a "direct" and "indirect" circuit within the basal ganglia. A minor change in its functioning results in severe neurological disorders. These parallel circuits diverge according to their nigrostriatal targets; the direct pathway is said to preferentially target striatal D1 receptors, whereas the indirect pathway is said to preferentially target D2 receptors [6]. These receptors are responsible to modulate excitation and inhibition in the circuit, respectively. Ultimately, both pathways project to the cortex via the thalamus (Figure 2: Basal Ganglia). Correct balance between the two pathways insures smooth running of the basal ganglia which in return insures proper functioning of voluntary motor control functions in the body.

To address the role the basal ganglia play in movement and movement disorders, neuro-computational model (2005) [6] constructed a novel computational model of the basal ganglia to address the role of the direct and hyper direct circuits in network function. We have used this computational model [7] and performed discrete modeling using [3].

Pathologically [8], Parkinson's disease is characterized by severe loss neurons in SNc called dopaminergic neurons which releases dopamine. Dopamine is a neurotransmitter which plays a significant role in many neurons. Loss of such cells is so severe that when a patient appears at the clinic with its sign and symptoms and before starting the treatment, it is estimated that approximately 60% to 70% of the SNc dopamine cells are lost [3]. The main features which results in such phenomena are the reduced impulses from dopaminergic neurons in SNn to the striatum (D2), increase in excitation of the STN and the globus pallidus internal, and more inhibition of the thalamus.

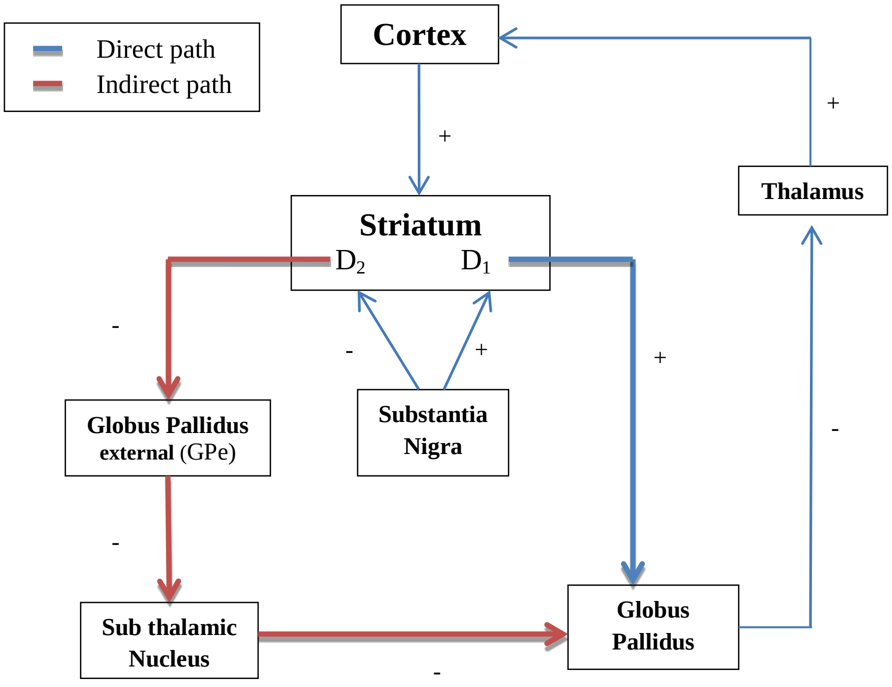


Figure 5: Basal Ganglia subdivided into five more components. Its shows its interaction with thalamus and cortex

1.3 Related Work

First computational model of basal ganglia was developed by Frank in 2001 [9] for better understanding of the interactions between cortex and basal ganglia in a working memory. In his work, he has analyzed the effect of difference parameters on learning and role of dopamine is highlighted showing that firing of striatal neurons at right time is very important for proper functioning of basal ganglia. In short, his paper presents the nature of basal ganglia that is in working memory when cortex is continuously firing over time however, the basal ganglia fires only at selected times and updates the memory states. The above computational model developed was used for further analysis of basal ganglia and implemented on different focus areas scenarios for example dynamic dopamine modulation (2004) [10], computational reinforcement learning perspective (2008)[11].

Another work by bradley Voytek [7], presented an abstract model of basal ganglia in which he oversimplified the model for more closer analysis of over view of different models. Every model has failed ultimately to prove the exact working of basal ganglia. Leblois et al. (2006) failed give an explanation for the importance of basal ganglia in cognitive processes. In frank [9] model, effect of depletion of dopaminergic neurons is analyzed but in next step (2006) [12], uses a model incorporating few interconnections within the nuclei hence different model used.

1.5 Contribution

For the first time, qualitative approach has been used for modeling and analysis of a biological neural network (BNN). Performing this type of modeling gives a new insight into the basal ganglia interactions and possible transitions. The BNN is represented as directed graphs where nodes represent neuron and edges represent the interaction between neurons. Basal ganglia is an essential biological neural network (BNN) consisting of five main regions namely Striatum (D1 and D2), thalamus, Globus Pallidus internal, Globus Pallidus external, Sub Thalamic Nucleus, Substantia Nigra and is strongly connected with cortex which is mainly responsible for motor functions in the body. Using René Thomas' qualitative model and GENOTECH, a Biological Regulatory Network tool (BRN), we constructed the directed graph of Basal Ganglia resulting in a state graph consisting of all possible states. The state graph resulted in one stable state and four cyclic trajectories. As the René Thomas' qualitative model is a total abstraction of time, therefore we incorporated the two types of delays that is activation and inhibition delay, developing a hybrid system. The activation delay is the synaptic delay caused during the stimulating and generation of action potential. The inhibition delay occurs between the firing and resting state of a neuron. Hybrid modeling of the same model helps us understand and analyze the system more closely by finding the constraint parameters due to which a specific dysfunction in basal ganglia occurs. For hybrid modeling, the state of the art, Hytech model checker is used resulting in state and path constraints which are delay constraints. By controlling these constraints, we can prevent the undesirable states and paths resulting in normal and healthy functioning of basal ganglia. Previously, mathematical modeling [24] and computational modeling [9] approaches have been carried out however, t it performs quantitative modeling of basal ganglia instead of qualitative.

1.6 Thesis Outline

Outline of the thesis is: Chapter 2 discussed the methodology used in the thesis. It explains in detail modeling of biological neural network (BNN) using qualitative and hybrid modeling and a short introduction to Hytech model checker. In chapter 3, modeling of basal ganglia using both modeling is presented with detailed description of complete model being used and its parameters. Chapter 4 presents the results and discussion of the thesis. Chapter 5 summarizes the basal ganglia model developed using qualitative modeling and how it can further contribute to the field of neuroscience.

CHAPTER 2: METHODOLOGY

2.1 Qualitative Modeling

Ordinary differential equations referring to the change in time of each quantity for example temperature or pressure are used for modeling of biological systems. As a system grows and interacts more, the complexity increases vigorously therefore it gets difficult to know the exact known quantity. In 1970, René Thomas' [13], introduced a qualitative approach for modeling biological regulatory networks based on Boolean logic [14,15]. The limitations of modeling Boolean logic is that is only has two states that is '0' and '1'which is not sufficient to tackle all problems. However, in later stages René Thomas' introduced another feature `Kinetic logic` which is closely approximates modeling based on ordinary differential equations [16, 17, 18, 19, 20, 21].In this modeling neurons are being considered as discrete states with 0 representing rest state while 1 represents firing state.

For Biological regulatory Networks (BRN's), different formalism have been Proposed among which differential equations (ODEs and PDEs) is an example.

Figure 3: Sigmoid curves (ODEs and PDEs based approach) shows sigmoid nature of ODEs and PDEs and Figure 4: Qualitative approach showing discrete approximation of sigmoid curves shows the discrete approximation of sigmoid curves.

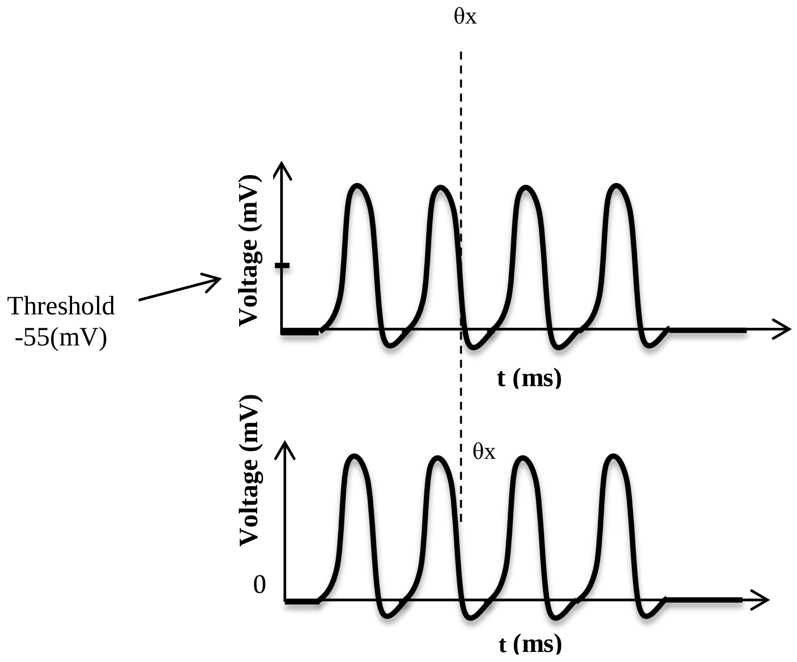


Figure 6: Sigmoid curves (ODEs and PDEs based approach)

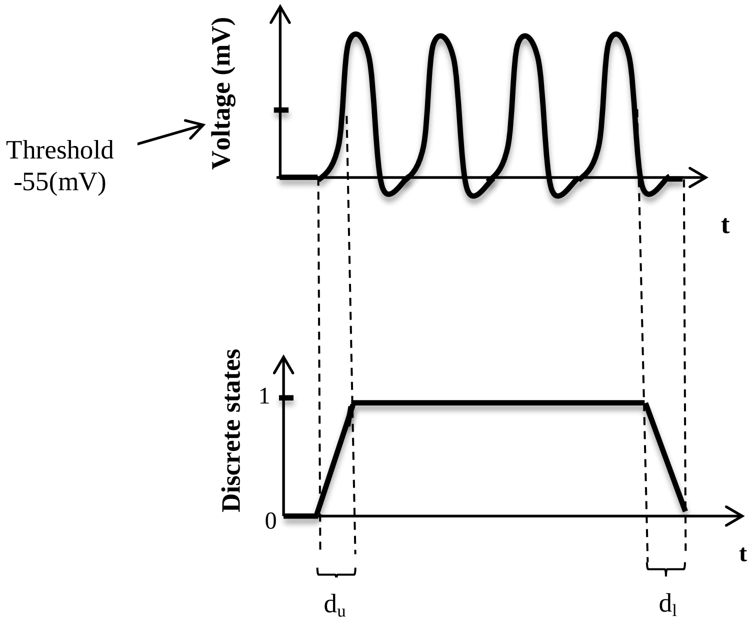


Figure 7: Qualitative approach showing discrete approximation of sigmoid curves

2.1.1 Discrete Modeling of Basal Ganglia associated BRN using GENOTECH

GENOTECH [22] and GINsim (Gene Interaction Network simulation) [23] are used for qualitative modeling tools for BRN with a simple graphical user interfaces (GUI). GENOTECH tool is based on Thomas' formalism. It takes two inputs from the user: BRN model as an input and its corresponding logical parameters form. It produces the whole state space by itself which may have one or more stable states, cycles and paths between any two states and can be easily identified. GINsim has similar features.

2.1.2 Modeling Equations

Through vigorous experiments with equations, the below equations showed the best results that is they were in accordance to the biologist results [5]. For modeling in GenoTech logic parameters are taken as input. These equations are based the Rene' Thomas. If a neuron has a negative ($-$) input i.e. it's inhibitory, it will prevent the target neuron from firing and will be in resting state. On the other hand, if a neuron has positive (+) input i.e. excitation so the target neuron will excite the target and results in an action potential. The following are the equations used for each location in basal ganglia and the system.

- K_{D2} = Cortex. $\overline{SNc}$
- K_{GPe} = $\overline{D2}$
- K_{GPi} = $\overline{GPe} + \overline{D1}$
- $K_{cor\text{-}thal}$ = $\overline{GPi}$
- K_{D1} = cortex . SNc
- K_{SNC} = 1

2.2 Hybrid Modeling

Hybrid modeling is the amalgamation of both discrete and continuous modeling with states being a discrete entity having continuous variables (clock). In Biological Neural Networks (BNN), the neurons are discrete states representing an instantaneous change between locations that is transfer of signal between two neurons [24] whereas the continuous transitions represents the time elapse while it’s in the respective location [25]. The discrete modeling above consists of only two states however in reality the basal ganglia (BNN) behaves in a nonlinear and continuous manner and this is where discrete modeling framework fails. In order to overcome such limitations, many proposals have been proposed [26] among which one proposed by Ahmad et al. [27] is hybrid modeling where sigmoidal nature evolutions are modeled with piecewise linear curves (see **Figure 3**). In this model concept of time interval and clocks is used.

The following table shows the comparison between biological networks and hybrid modeling [28]:

Biological Network	Hybrid Modeling
Reaches a specific value (threshold)	Discrete events
Kinetic methodology is being used	Different Equations are used
It has gene to gene interactions	It has concurrency
Its behavior is stochastic	It depends on random numbers

Table 1: Difference Between biological and Hybrid modeling

2.2.1 Approach of Rene` Thomas

According to Rene` Thomas [17], a directed graph i.e. B = (N, $\mathcal{A}$), a node n ϵ N may have a number of predecessor and successors B^- (n) and B^+(n) respectively.

Definition 4. *[29] A Biological regulatory network is a tuple B = (N, $\mathcal{A}$, f, g, t, Z) where*

- *(N, $\mathcal{A}$) is a directed graph denoted by B,*
- *F is a function from $\mathcal{A}$ to $\mathbb{N}$*
- *g is a function from $\mathcal{A}$ to $\{+, -\}$*
- *t is a function from $\mathcal{A}$ to $\mathbb{N}$ such that, for all x ϵ N, if $B^+(x)$ is not an empty then t(x, n)| n ϵ B^+ (x) = 1……., 1(u)*
- *Z = Z_n |n ϵ N is a set of maps: for each n ϵ N, Z_n is a function from 2^{B-} to 0,…..f(n) such that $Z_n(w) \leq Z_n$ (w`) for all w $\subseteq$ w` $\subseteq$ B^- (n)*

The map l describes the domain of each variable n: if f(n) = k, the abstract concentration on n holds its value in 0,1…,k. Similarly, the map g represents the sign of the generation (+ is for an activation , - is for inhibition). t(x, n) is the threshold of the regulation from x to n: this regulation takes place if and only if the abstract concentration of u is above t(x,n), in such a case the regulation is said active. The condition on these thresholds states that each variation of the level of x induces a modification of the set of active regulations starting from x. For all x $\subseteq$ [0,……f(x)$-$ 1], the set of active regulations of u, when the discrete expression level of x is u, differs from the set when the discrete expression level is u + 1. Finally, the map Z_n allows us to define, what is the effect of a set of regulators on the specific target n. If this set is w $\subseteq$ B^-(x), then, the target x is subject to a set of regulations which makes it to evolve towards a particular level Z_n (w).

Defination 5: *(States). [29] A state μ of a BRN $B = (N, \mathcal{A}, f, g, t, Z)$ is a function from N to $\mathbb{N}$ such that μ (x) ϵ {0,…..f(x)} for all variables n ϵ N. We denote E^B the set of states of B. When μ (x) $\geq$ t(x, n) and g(x,n) = +, we say x is a resource of n since the activation takes place. Similarly when $\mu(x) < t(x, n)$ and g(x, n) = $-$, x is also a resource of n since the inhibition does not take place (the absence of the inhibition is treated as an activation).*

Definition 6: *(Resource Function). [29] Let $B = (N, \mathcal{A}, f, g, t, Z)$ be a BRN. For each (n ϵ N we define the resource function $w_n : E^B \rightarrow 2^{B-(n)}$ by: w(μ) = { μ = $B^{-}(n)$|(μ(n)) $\geq$ t(x, n) and s(x,n) = + or μ(n)) $\geq$ t(x, n) and s(x,n) = $-$ })*

As said before, at state μ, $Z_n (w_n (\mu))$ gives the level towards which the variable n tends to evolve. There are three considerable cases,

- *if $\mu(n) < Z_n (w_n (\mu)$ then n can increase by one unit*
- *if $\mu(n) > Z_n (w_n (\mu)$ then n can decrease by one unit*
- *if $\mu(n) = Z_n (w_n (\mu)$ then n cannot evolve.*

2.3 Linear Hybrid Automata

A hybrid automaton is a mathematical model for hybrid systems, which in a single formalism combines captures both discrete and continuous, with interactions (transition) and differential equations are the changes respectively.

Definition 1.1 *[Hybrid automata] [30, 31, 32] A hybrid automaton H consists of the following components.*

Variables: A finite set Y= f {x_1, …..x_n} of real-numbered variables. The number n is called the dimension of N. We write X for the set f{x_1,…., x} of dotted variables (which represent first derivatives during continuous change), and we write X` for the set f{x`,…., x`$_n$} of primed variables (which represent values at the conclusion of discrete change).

Control graph. A finite directed multi-graph (N,E). The vertices in N are called control modes. The edges in E are called control switches.

Initial, invariant, and flow conditions. Three vertex labeling functions init , inv, and flow that assign to each control mode n ϵ N three predicates. Each initial condition init(n) is a predicate whose free variables are from N. Each invariant condition inv(n) is a predicate whose free variables are from X. Each flow condition flow(n) is a predicate whose free variables are from N ∪ N.

Jump conditions. An edge labeling function jump that assigns to each control switch e ϵ E, Each jump condition jump(e) is a predicate whose free variables are from X ∪ X`.

Events. A finite set Σ of events, and an edge labeling function event : E → Σ
that assigns to each control switch an event.

2.4 Model Checking

Model Checking is a verification technique that explores all possible in the state space. Variety of software's tools called model checker are used in model checking, each implemented on different modeling approach. Model Checker in a systematic manner examines all possible states of the system and produces a state graph. However they have few limitations that is it can handle state space of about 108 to 109 with explicit state space enumeration. In past few years, due to the vigorous growth in the computational power, biologists are able to produce more efficient and fast methods for analyzing BNNs.

2.4.1 HyTech Model Checker

HyTech [33, 34] is a tool for automated analysis of hybrid systems by computing with polyhedral state sets. Key features of HyTech which aids in designing of hybrid systems is that it checks system requirements i.e. reachability property and its ability to perform parametric analysis, i.e., to determine the values of design parameters for which a linear hybrid automaton satisfies a temporal-logic requirement which is by symbolic calculations.

HyTech input consists of the following two parts:

1. System description: In this part user has to give textual description of its linear hybrid automata. HyTech has its own descriptive language
2. Sequence of analysis commands: This is where user has to give its input regarding the sequence on input and analyze the system

CHAPTER 3: MODELING OF BASAL GANGLIA

3.1 Modeling approach

3.1.1 Discrete modeling of Basal Ganglia

The qualitative modeling of basal ganglia is implemented in GENOTECH (as shown in (2.1.1 Discrete Modeling of Basal Ganglia associated BRN using GENOTECH). The timing diagram of one normal path is shown in Figure 11. In the thesis, GENOTECH is used however GINsim can be used instead. In order to use such a tool, the following steps are taken:

1. Construct BRN labeled directed graph: GUI of GENOTECH is user friendly interface. Right click the interface provides us with two drop menus to create and edit BRN nodes and interactions respectively. Along with creating the biological entities, the user has to enter a set of logical parameters for each entity according the interactions and research data (see Figure 5: Basal Ganglia model in GENOTECH and Figure 7: An example setting logical parameters for an entity in GENOTECH).
2. Generation of a state graph: After the constructing the BRN, the next step is to generate a state graph for the respective model. This can be done by using the command 'state graph' in the File menu. It appears in a new window with its own set of commands which makes analysis easier. The GUI of state graph represents the stable states in red. To find cycles click the 'cycles' drop menu from the right and observer the cycle in GUI interface. Same procedure for finding the paths between

two states and neighboring states (see Figure 6: State graph of basal ganglia using GENOTECH with all cycles appearing in the right bottom window and options of show path, show cycles and show neighboring cycles available at the top right of the state graph window. Steady state is represented by red color.). In case, the state graph is large and difficult to analyze in state graph window, GENOTECH also provides an option to save the state graph in DOT format [35] for visualization in Graphviz tool [36].

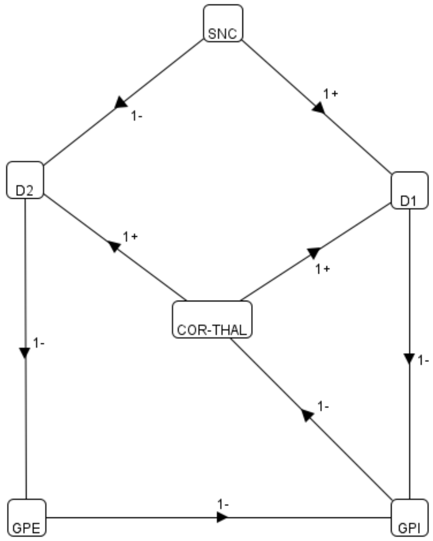
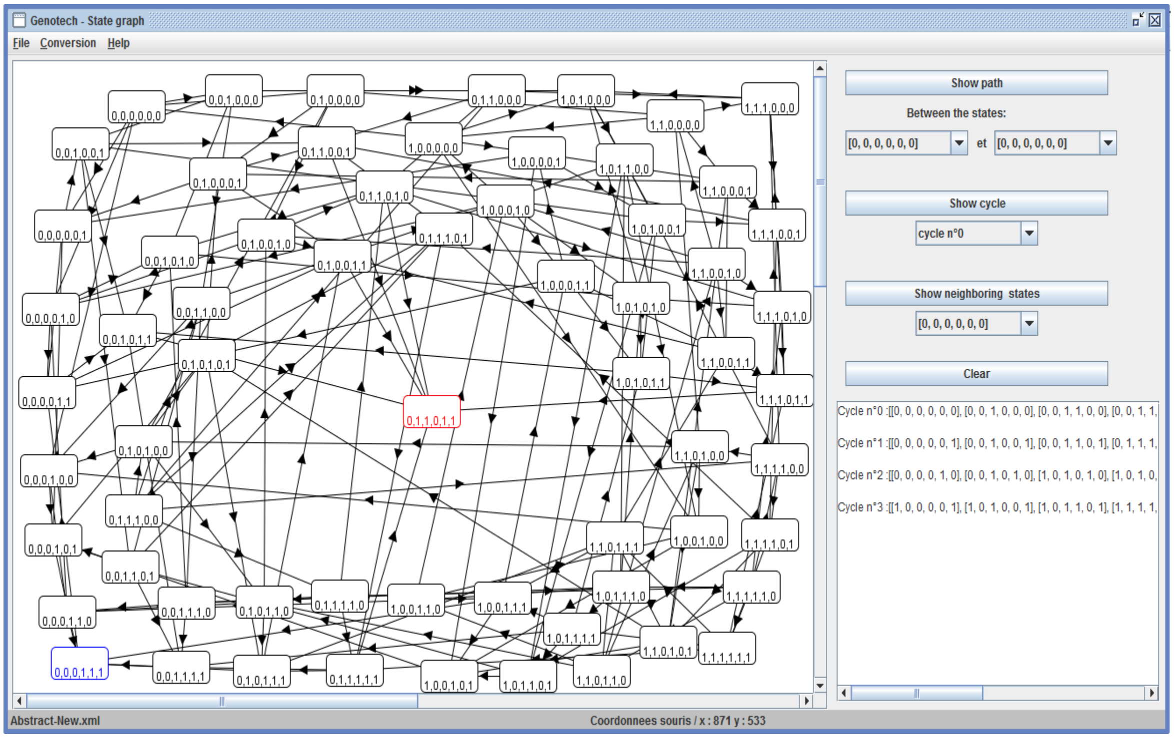


Figure 8: Basal Ganglia model in GENOTECH



Between the states:

[0, 0, 0, 0, 0, 0] et [0, 0, 0, 0, 0, 0]

cycle n°0

[0, 0, 0, 0, 0, 0]

```
Cycle n°0 :[[0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 1, 1, 0, 0], [0, 0, 1, 1,
Cycle n°1 :[[0, 0, 0, 0, 0, 1], [0, 0, 1, 0, 0, 1], [0, 0, 1, 1, 0, 1], [0, 1, 1, 1,
Cycle n°2 :[[0, 0, 0, 0, 1, 0], [0, 0, 1, 0, 1, 0], [1, 0, 1, 0, 1, 0], [1, 0, 1, 0,
Cycle n°3 :[[1, 0, 0, 0, 0, 1], [1, 0, 1, 0, 0, 1], [1, 0, 1, 1, 0, 1], [1, 1, 1, 1,
```

Abstract-New.xml

Coordonnees souris / x : 871 y : 533

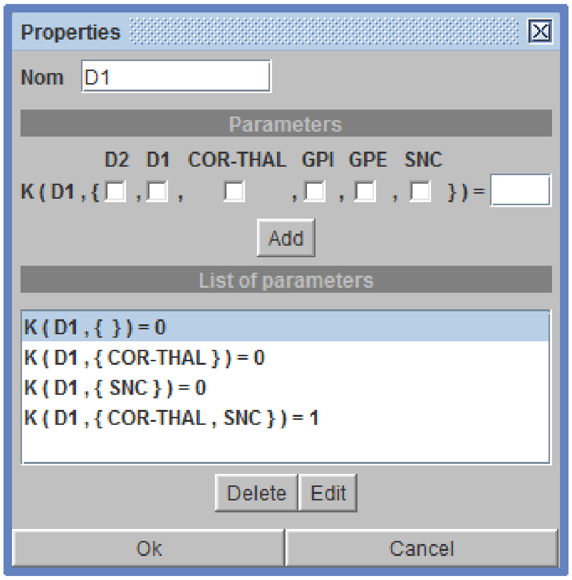
Figure 9: State graph of basal ganglia using GENOTECH with all cycles appearing in the right bottom window and options of show path, show cycles and show neighboring cycles available at the top right of the state graph window. Steady state is represented by red color.

3.1.3 Modeling parameters

Change from one state to another depends on the logical parameters allotted to each biological entity. At any instant the evolution of level α depends only on set of activators denoted by $w(Z_\alpha)$. By logical parameters, the effect of $w(Z_\alpha)$ on the evolution is Z_α can be known.

The logical parameters correspond to the level towards which gene α evolves:

1. if $\mu(n) < Z_n (w_n (\mu)$ then n can increase by one unit
2. if $\mu(n) > Z_n (w_n (\mu)$ then n can decrease by one unit
3. if $\mu(n) = Z_n (w_n (\mu)$ then n cannot evolve



Properties
Parameters
D2 D1 COR-THAL GPI GPE SNC
K (D1 , { □ , □ , □ , □ , □ , □ }) =
Add
List of parameters
Delete
Edit
Ok
Cancel

Nom D1

K (D1 , { }) = 0
K (D1 , { COR-THAL }) = 0
K (D1 , { SNC }) = 0
K (D1 , { COR-THAL , SNC }) = 1

Figure 10: An example setting logical parameters for an entity in GENOTECH. The first row shows the order used in qualitative modeling

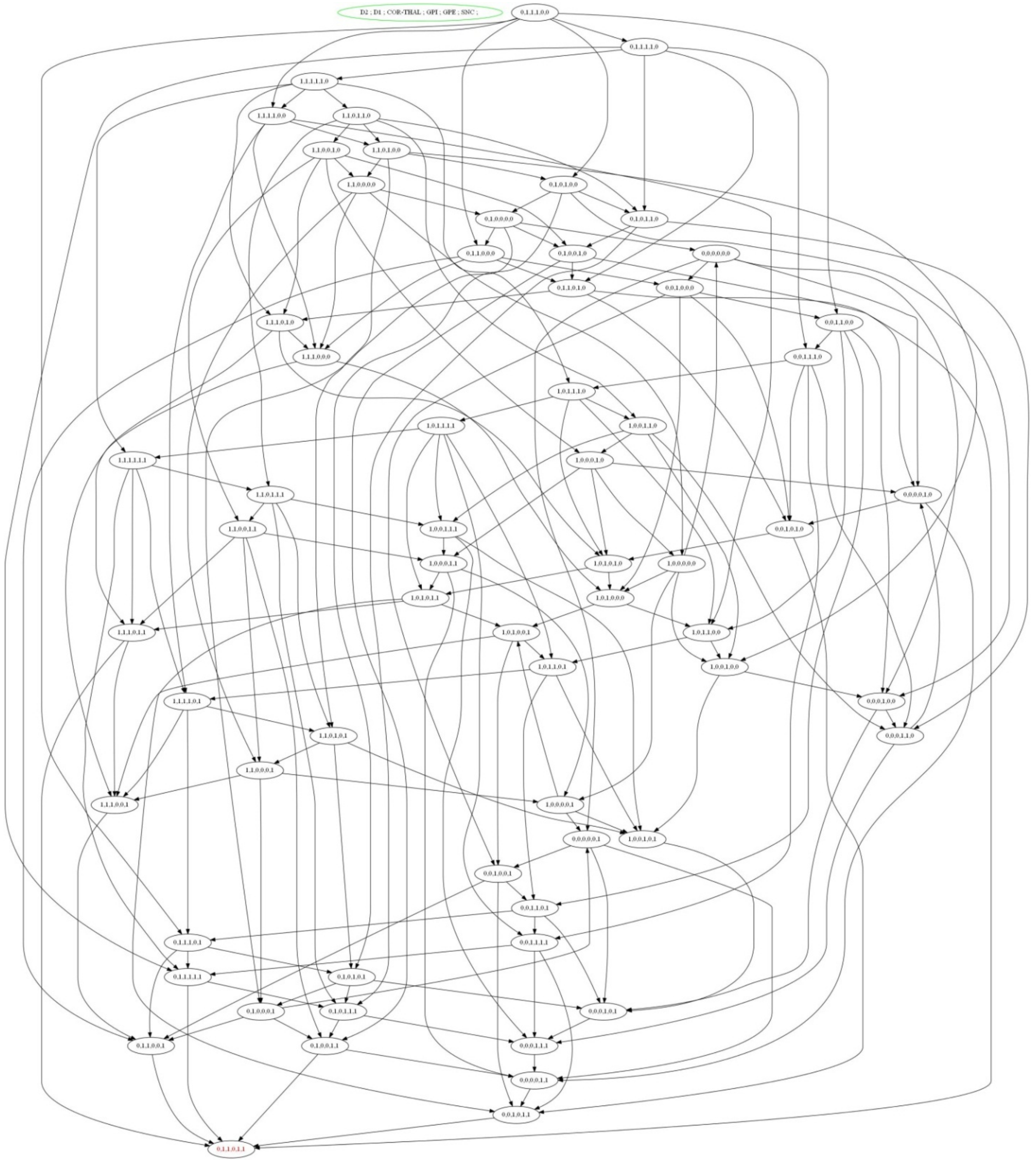


Figure 11: State graph of basal ganglia using GenoTech with respective logical parameters. It represents the state graph with (D_2, D_1, cor_thal, GPi, GPe, SNc) order

3.2 Hybrid Modeling

In order to enhance our discrete modeling, we performed hybrid analysis for finding delay constraints of this model, using HyTech model checker [33]. The major reason for using HyTech as a hybrid model checking tool is that it is rich in parametric analysis [37].

In our model, we have incorporated are two delays. One being the synaptic delay (ds) (see Figure 9) is caused between the stimulation of the neuron and the action potential being actually fired. This is the major cause of many neurodegenerative diseases including Alzheimer and Parkinson's disease. Synaptic delay is delay is defined as ds. The other type of delay is constant delay (see Figure 9) which is after the action potential is fired and reaches back to rest state again. This delay is constant for every neuron during which no other action potential can be fired. As the dc delay is constant therefore we are incorporating only one delay that is synaptic delay (ds).

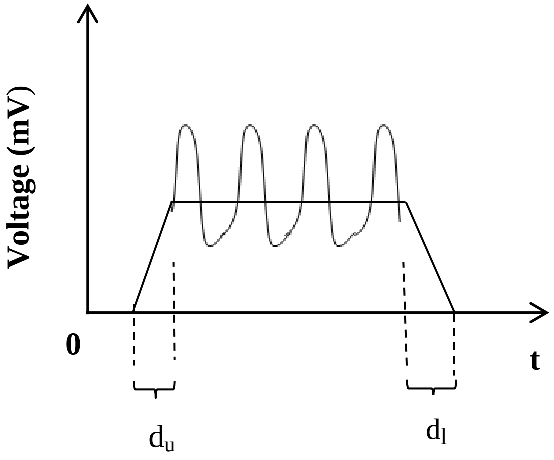


Figure 12: Types of delays incorporated in a single neuron using HyTech basal ganglia model

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Discrete Modeling using GENOTECH

Implementing the model in GenoTech resulted in four cycles (see Table 2: The cycles obtained from GENOTECH) and one steady state. Basal Ganglia have various functions among which controlling body movements (motor function) [38] is a major one. In order to control movements, thalamus is the focal point which further controls the cortex and hence the motor functions [39].The cycle in (Figure 10: Cycle 0: Resting cycle i.e. Inhibition cycle with low dopamine8) represents the **resting** cycle of basal ganglia. In resting state, there is no stimulus for muscle movement therefore the function of basal ganglia is to continuously inhibit the thalamus [3] until no excitation or increase in dopamine level occurs. In this state, no dopamine is released and hence STN =0 [3].

Cycle n°0 :	[0, 0, 0, 0, 0, 0] → [0, 0, 1, 0, 0, 0] → [0, 0, 1, 1, 0, 0] → [0, 0, 1, 1, 1, 0] → [1, 0, 1, 1, 1, 0] → [1, 0, 0, 1, 1, 0] → [1, 0, 0, 0, 1, 0] → [1, 0, 0, 0, 0, 0] → [0, 0, 0, 0, 0, 0]
Cycle n°1	[0, 0, 0, 0, 0, 1] → [0, 0, 1, 0, 0, 1] → [0, 0, 1, 1, 0, 1] →[0, 1, 1, 1, 0, 1]→ [0, 1, 0, 1, 0, 1] → [0, 1, 0, 0, 0, 1] → [0, 0, 0, 0, 0, 1]

Cycle n°2 :	[0, 0, 0, 0, 1, 0] → [0, 0, 1, 0, 1, 0] → [1, 0, 1, 0, 1, 0] → [1, 0, 1, 0, 0, 0] → [1, 0, 1, 1, 0, 0] → [1, 0, 0, 1, 0, 0] → [0, 0, 0, 1, 0, 0] → [0, 0, 0, 1, 1, 0] → [0, 0, 0, 0, 1, 0]
Cycle n°3 :	[1, 0, 0, 0, 0, 1] → [1, 0, 1, 0, 0, 1] → [1, 0, 1, 1, 0, 1], [1, 1, 1, 1, 0, 1] → [1, 1, 0, 1, 0, 1] → [1, 1, 0, 0, 0, 1] → [1, 0, 0, 0, 0, 1]

Table 2: The cycles obtained from GENOTECH

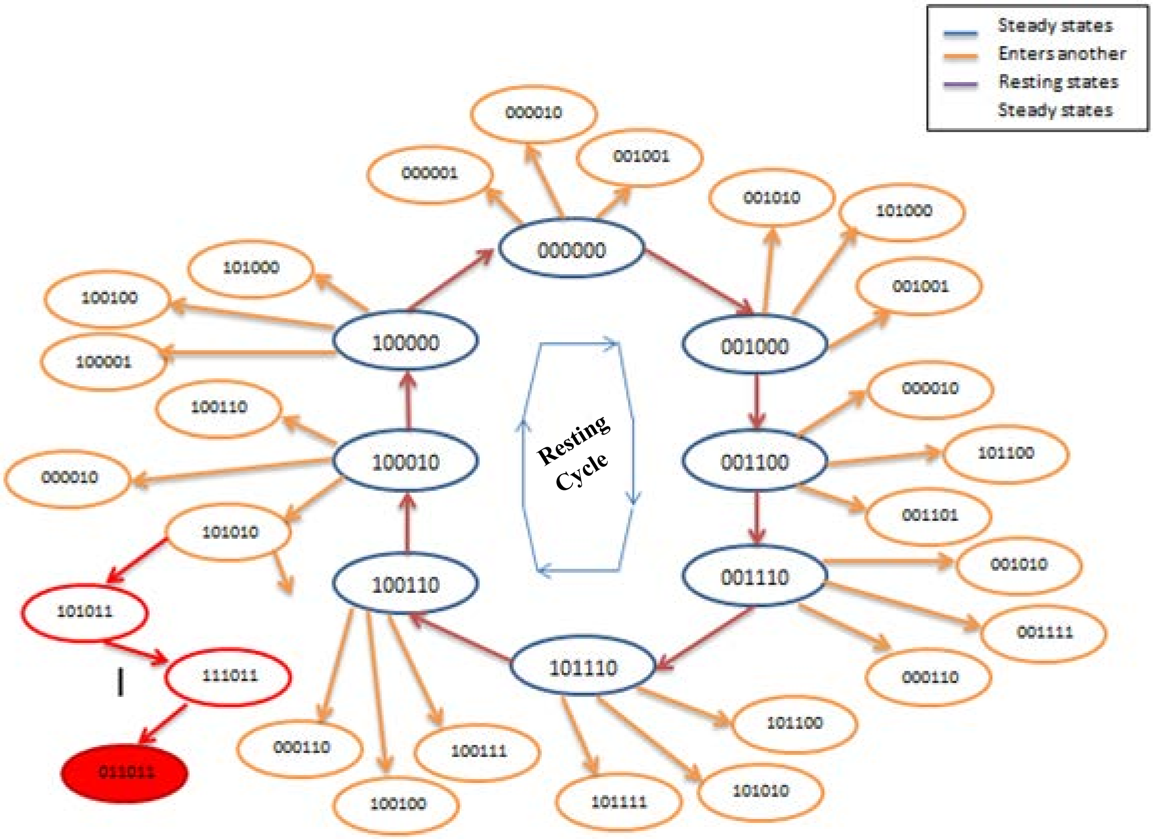


Figure 13: Cycle 0: Resting cycle i.e. Inhibition cycle with low dopamine

Within normal basal ganglia, when excitation of cortex due to stimulus from other part of the brain, dopamine is immediately released and transition from cycle 0 (Figure 10: Cycle 0: Resting cycle i.e. Inhibition cycle with low dopamine) to cycle 1 (Figure 11: Cycle 1: Excitation with high Dopamine level) occurs. This causes the excitation of target areas. In order to stop it, inhibition of cortex must happen be stopped for which it should enter cycle 3 with dopamine still being released in the body.

Transition to cycle 3 (Figure 13: Cycle 3: Inhibition with high level of dopamine) must only occur from location 011101 within cycle 2 otherwise it enters the steady state 011011 which means continuous excitation of cortex and tremors occur. In order to avoid such scenario, proper functioning of cortex and GPe is important [3]. Loss or death of neurons in GPe means more excitation to cortex and more reliable to enter the steady state. External perturbation must occur for the tremors to stop and enter the resting cycle.

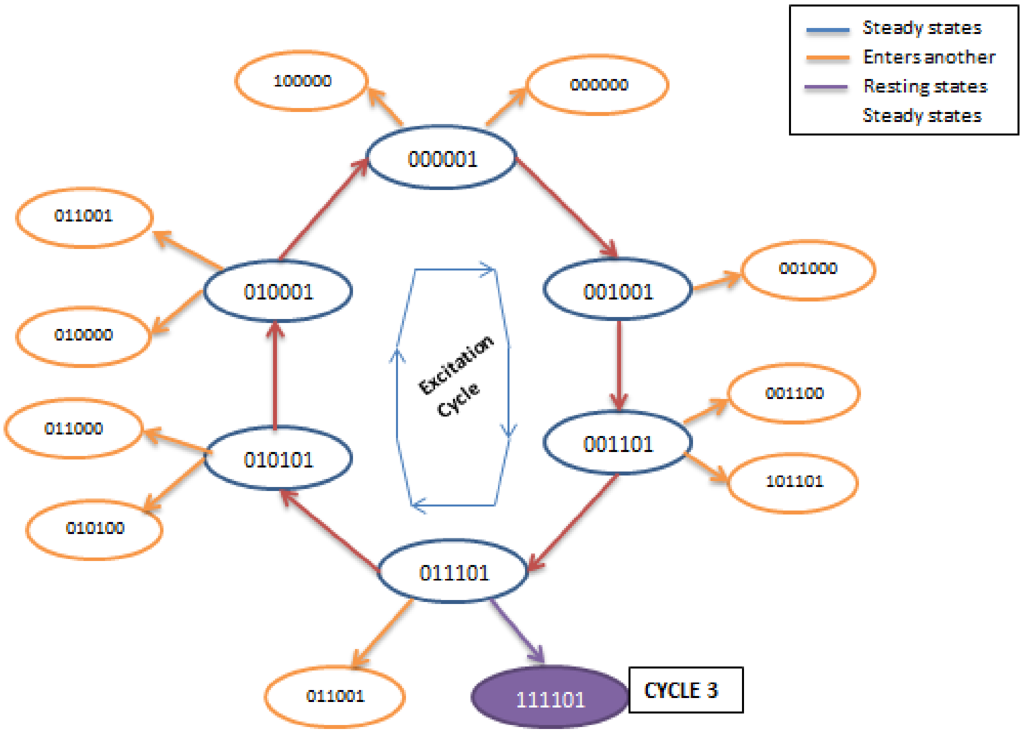


Figure 14: Cycle 1: Excitation with high Dopamine level

Basal Ganglia is extremely sensitive to minute change in organizational or structural may lead to miscommunication between thalamus and cortex resulting in a disease states like parkinsonism or Huntington’s disease. If the indirect path is dominating the direct path of basal ganglia as it is in Figure 12: Cycle 2: Inhibition when low level of dopamine with low dopamine level, may result in loss of the ability to detect the input signal from the stimulating area and hence lose the ability to communicate this information further to the respective target area. This is what happens in Parkinson and how patients suffering from such disease have slowed movements (bradykinesia).

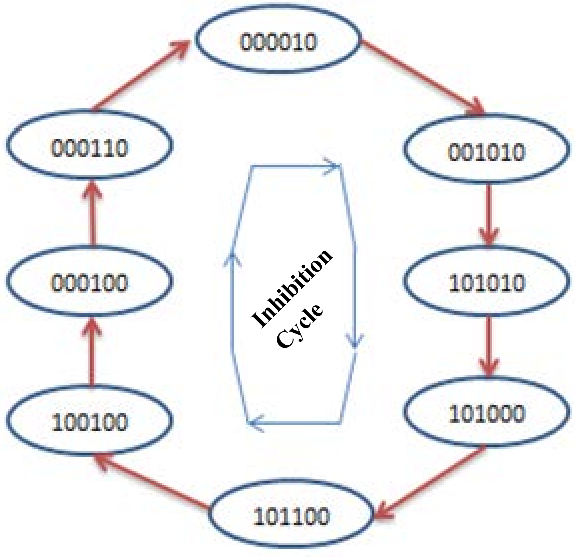


Figure 15: Cycle 2: Inhibition when low level of dopamine

The cycle below is a normal inhibition cycle with high dopamine level. This cycle bring the system back to normal by inhibiting the excitation and then finally enters the resting cycle when STN=0 which means no dopamine is released as no excitation required of cortex needed.

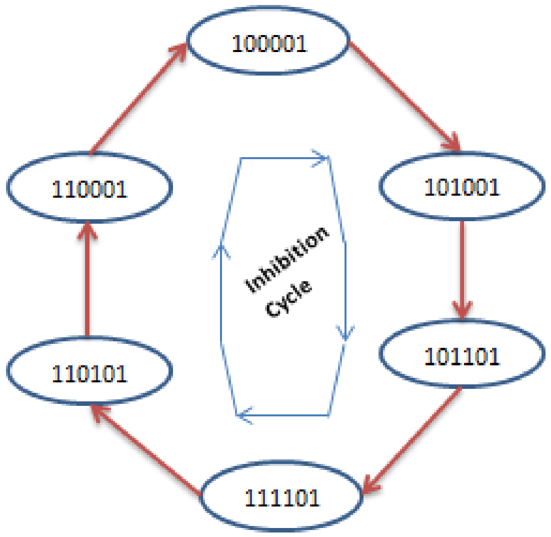


Figure 16: Cycle 3: Inhibition with high level of dopamine

The results are explained in detail by the timing diagrams (Figure 15 & Figure 16)

As you can see in normal resting cycle, there is no steady state and it will stay in the respective cycle until there is a stimulus or disturbance due to various factors may result in a disease state (Figure 15). The change due to which the resting cycle ends up in a tremor state is because of faster rate of cortex as compared to that of GPe at location 100010. In steady state, the cortex is suddenly excited and remains excited for some time, resulting in a steady state i.e. tremors.

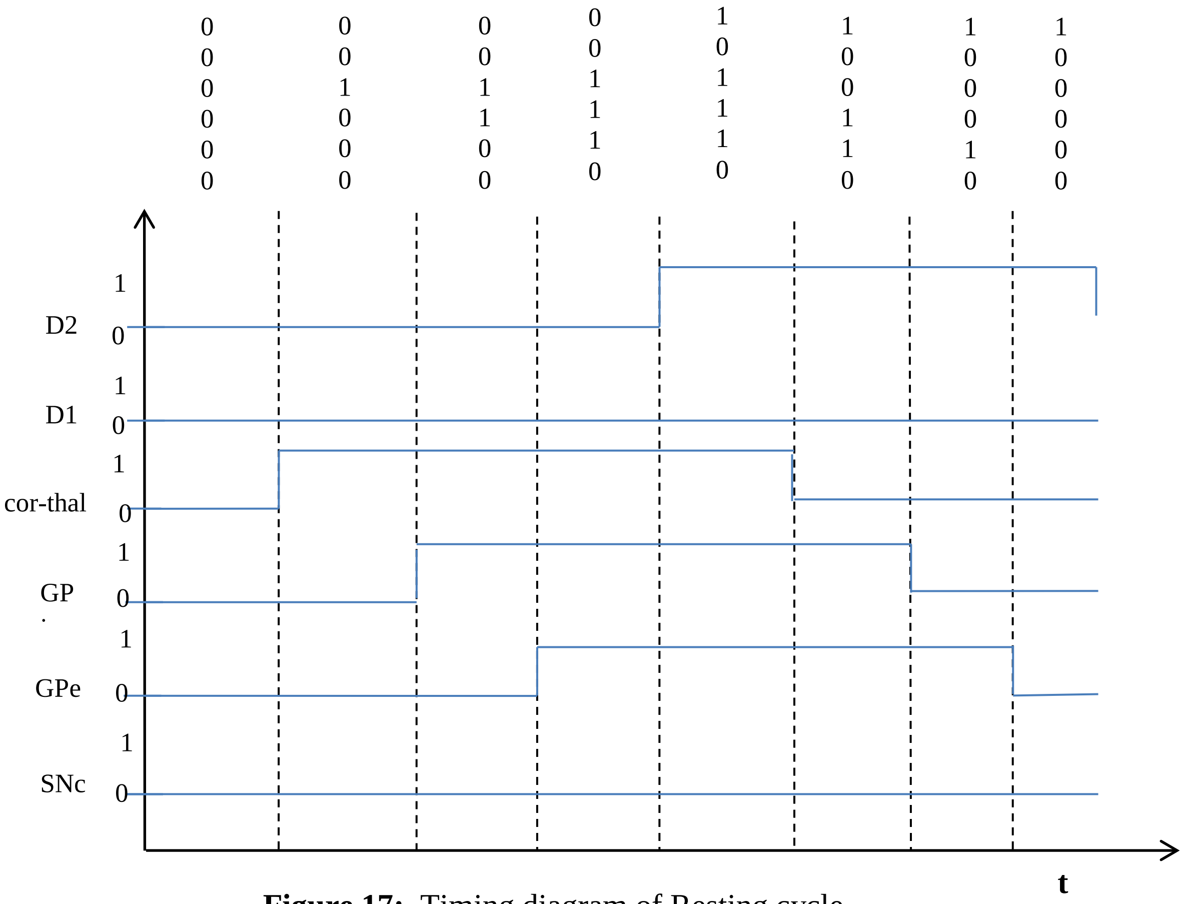


Figure 17: Timing diagram of Resting cycle

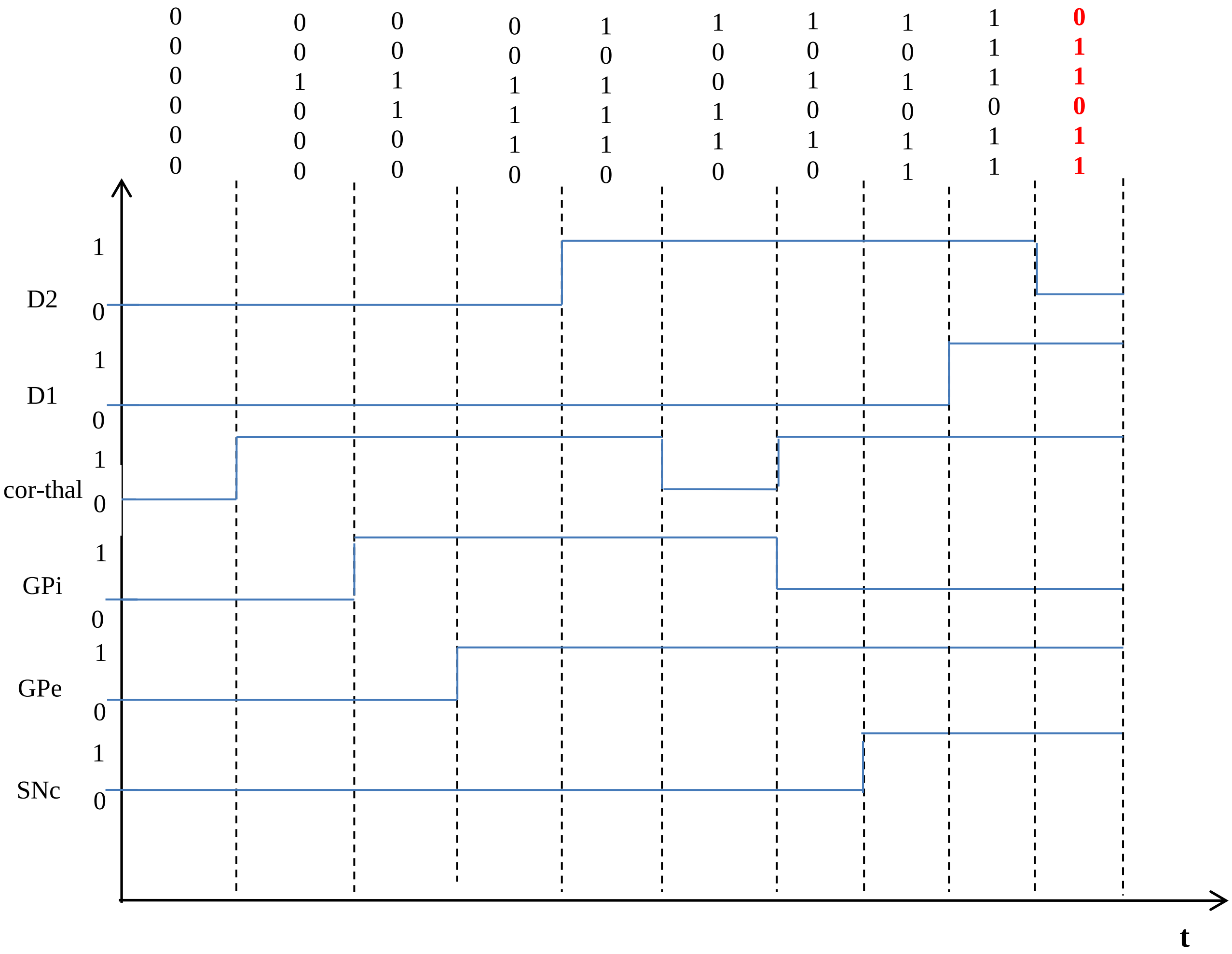


Figure 18: Timing diagram of steady state (in red)

4.2 Hybrid Modeling using HyTech

By incorporating the synaptic delay (Figure 9) in the basal ganglia model helps us in finding the delay constraints for the respective states. In our analysis, we have implemented this modeling on the resting cycle (Figure 10) of basal ganglia as detected in the discrete modeling and using HyTech (Figure 17) we calculated the conditions that is delay constraints which would prevent the system from going into disease state which causes oscillations (tremors). Delay constraints of the model with transition states are shown in Table 3.

Transition States	**Delay Constraints**
000000 → 001000	$d^{+}_{\text{cor-thal\_u}} \leq d^{+}_{\text{SNc\_u}} + d^{+}_{\text{GPe\_u}} + d^{+}_{\text{GPi\_u}}$
001000 → 001100	$d^{+}_{\text{GPi\_u}} \leq d^{+}_{\text{SNc\_u}} + d^{+}_{\text{GPe\_u}} + d^{+}_{\text{D2\_u}}$
001100 → 001110	$d^{+}_{\text{GPe\_u}} \leq d^{-}_{\text{cor\_thal\_l}} + d^{+}_{\text{SNc\_u}} + d^{+}_{\text{D2\_u}}$
001110 → 101110	$d^{-}_{\text{D2\_l}} \leq d^{+}_{\text{SNc\_u}} + d^{-}_{\text{GPi\_l}} + d^{-}_{\text{cor\_thal\_l}}$
101110 → 100110	$d^{-}_{\text{cor\_thal\_l}} \leq d^{+}_{\text{SNc\_u}} + d^{-}_{\text{GPe\_l}} + d^{-}_{\text{GPi\_l}}$
100110 → 100010	$d^{-}_{\text{GPi\_l}} \leq d^{+}_{\text{SNc\_u}} + d^{-}_{\text{D2\_l}} + d^{-}_{\text{GPe\_l}}$
100010 → 100000	$d^{-}_{\text{GPe\_l}} \leq d^{+}_{\text{SNc\_u}} + d^{-}_{\text{D2\_l}} + d^{+}_{\text{cor-thal\_u}}$
100000 → 000000	$d^{-}_{\text{D2\_l}} \leq d^{+}_{\text{SNc\_u}} + d^{+}_{\text{GPi\_u}} + d^{+}_{\text{cor-thal\_u}}$

Table 3: Delay constraints generated using HyTech for resting state i.e. cycle 0

In order to get the maximal efficiency out of hytech model checker, we calculated the path constraint (Table 4) for the cycle 0 which is the resting state. All path constraints for cycle 0 must be maintained in order to stay in the required state. For example the delay in lowering the signal (1 → 0) from cor_thal should be less than GPe (1 → 0) for it remain in the stable state. In case, cor_thal inhibits quickly as compared to GPe along with a continuous signal from will cause the system to enter the steady state using the path below.

100010 → 101010 → 101011 → 111011 → 011011

Figure 19: Showing the pathway which enters into a deadlock state

	Path Constraints
1.	$d_{\text{cor-thal\_u}} = 0$
2.	$d_{\text{GPe\_u}} + d_{\text{GPi\_l}} = d_{\text{D2\_l}} + d_{\text{GPe\_l}}$
3.	$d_{\text{D2\_l}} + d_{\text{GPi\_u}} + d_{\text{cor\_thal\_l}} = d_{\text{GPe\_u}} + d_{\text{GPi\_l}}$
4.	$d_{\text{GPe\_l}} \leq d_{\text{GPi\_l}}$
5.	$d_{\text{GPe\_u}} + d_{\text{GPi\_l}} \leq d_{\text{GPi\_u}} + d_{\text{cor\_thal\_l}} + d_{\text{GPe\_l}}$
6.	$d_{\text{GPi\_u}} + d_{\text{cor\_thal\_l}} \leq d_{\text{GPe\_u}} + d_{\text{GPi\_l}}$
7.	$d_{\text{GPi\_u}} \leq d_{\text{GPe\_u}}$
8.	$d_{\text{GPi\_u}} \geq 0$
9.	$d_{\text{GPe\_u}} + d_{\text{GPi\_l}} \leq d_{\text{SNc\_u}}$

Table 4: Path constraints for the cycle 0. Any one of the above conditions violated results in evolution of path away from cycle 0 and enters into another cycle and in specific conditions may end up in a steady state (disease state)

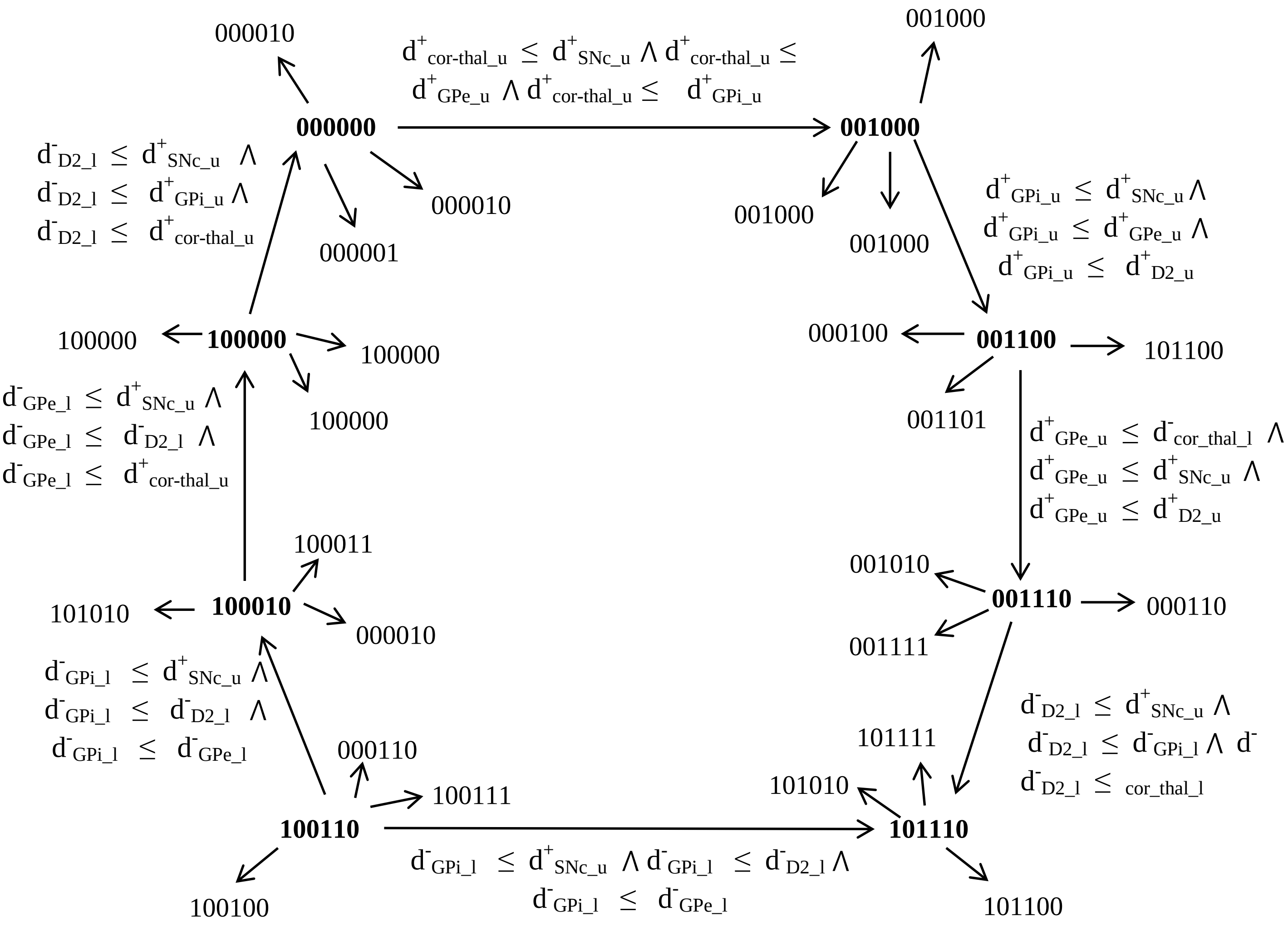

000010
$d^{+}_{cor\text{-}thal\_u} \leq d^{+}_{SNc\_u} \wedge d^{+}_{cor\text{-}thal\_u} \leq$
$d^{+}_{GPe\_u} \wedge d^{+}_{cor\text{-}thal\_u} \leq d^{+}_{GPi\_u}$
001000
000000
001000
$d^{-}_{D2\_l} \leq d^{+}_{SNc\_u} \wedge$
$d^{-}_{D2\_l} \leq d^{+}_{GPi\_u} \wedge$
$d^{-}_{D2\_l} \leq d^{+}_{cor\text{-}thal\_u}$
000010
000001
001000
001000
$d^{+}_{GPi\_u} \leq d^{+}_{SNc\_u} \wedge$
$d^{+}_{GPi\_u} \leq d^{+}_{GPe\_u} \wedge$
$d^{+}_{GPi\_u} \leq d^{+}_{D2\_u}$
100000
100000
100000
100000
000100
001100
101100
001101
$d^{-}_{GPe\_l} \leq d^{+}_{SNc\_u} \wedge$
$d^{-}_{GPe\_l} \leq d^{-}_{D2\_l} \wedge$
$d^{-}_{GPe\_l} \leq d^{+}_{cor\text{-}thal\_u}$
$d^{+}_{GPe\_u} \leq d^{-}_{cor\_thal\_l} \wedge$
$d^{+}_{GPe\_u} \leq d^{+}_{SNc\_u} \wedge$
$d^{+}_{GPe\_u} \leq d^{+}_{D2\_u}$
100011
101010
100010
000010
001010
001110
000110
001111
$d^{-}_{GPi\_l} \leq d^{+}_{SNc\_u} \wedge$
$d^{-}_{GPi\_l} \leq d^{-}_{D2\_l} \wedge$
$d^{-}_{GPi\_l} \leq d^{-}_{GPe\_l}$
$d^{-}_{D2\_l} \leq d^{+}_{SNc\_u} \wedge$
$d^{-}_{D2\_l} \leq d^{-}_{GPi\_l} \wedge d^{-}$
$d^{-}_{D2\_l} \leq {}_{cor\_thal\_l}$
000110
100111
101111
101010
100110
101110
$d^{-}_{GPi\_l} \leq d^{+}_{SNc\_u} \wedge d^{-}_{GPi\_l} \leq d^{-}_{D2\_l} \wedge$
$d^{-}_{GPi\_l} \leq d^{-}_{GPe\_l}$
100100
101100

Figure 20: Delay constraints of the cycle 0

4.3 Limitations of our method

Major disadvantage of model checking is state explosion problem. The state space of a system can be very large or even infinite therefore it is impossible to explore the entire state space. For model checkers, memory is more critical resource than time. In discrete modeling we have only two states that are 0 and 1 representing resting state and firing state respectively due to which our model is limited to two such behaviors. As the system is very complex, increasing the number of states resulted in state explosion problem using Hytech model checker. Another limitation of our model is the number of locations and transitions in discrete modeling using GenoTech. Increasing the number of locations to ten or more in GenoTech model checker, fail to give results for which we have to either use another software or use parallel programming and implement this using supercomputer.

CHAPTER 5: CONCLUSION AND FUTURE WORK

Basal ganglia play a very important role in Parkinson’s disease and Alzheimer’s disease. Both diseases occur due to a minute damage in basal ganglia. One of the symptoms of Parkinson’s disease is resting tremors [40]. In resting tremor, In our results we have detected the resting tremor. When the system is in resting cycle, it represents the systems resting state but sudden delay or change in behavior of the system at location 100010 to 101010 instead of 100000, the system may enter to the a particular pathway [Figure 19] resulting in resting tremors (deadlock state). The timing diagram (Figure 15: Timing diagram of steady state) shows how resting tremors occurs compared to the normal working of basal ganglia (Figure 14: Timing diagram of Resting cycle). When signal from cortex is continuous and its rate that is transition from 0 to 1 is high as compared to rate of change of GPe (transition from 1 to 0). Reason for low rate of GPe may be as following:

1. Loss of Neurons between D2 and GPe region [3,41]
2. Problem in Cortex region [41]

The results produced using qualitative modeling along with hybrid modeling in the thesis shows the strength of modeling with these techniques and how model checking results are in closely proximate with real Biological Neural Network of basal ganglia.

Basal ganglia is a very complicated system having dependency on various other parameters like learning through synaptic plasticity, death of neurons and many more. In future, these parameters can be incorporated into the hybrid model to make it a more realistic more and analyze the working of basal ganglia deeply with the help of state of the art model checkering tools.

Hytech results in delay constraints for each cycle. Minute delay in any of the constraints results in violation of normal behavior, resulting in excitation, inhibition or enters the disease state (011011). This disease state means the cortex is continuously excited causing tremors in real system. In order to stop tremors, an external input is required. This external signal either excites (D_2 , GPi) or inhibits (D_1, Cor_thal, GPe, SNc) due to which it enters one of the four cycles and again maintains a normal behavior.

References

[1] Michael Reed, “World Health Organization (WHO) Public health Challenges chapter 2” presented *at Scott Rich, Trinity ’12,* Research Training program in Biology

[2] Pei Ye, Emilia Entchevay, Scott A. Smolka, Radu Grosu, “Symbolic Analysis of the Neuron Action Potential” presented at *Bioinformatics and Biomedical Engineering*, 2008. ICBBE 2008, 836 - 839

[3] Dale Purves, George J. Augustine, David Fitzpatrick, William C.Hall, Anthony-Samiel Lamantia, James O. McNamara, S. Mark Williams, Neuroscience edition 3 chapter

[4] María-Trinidad Herrero, Carlos Barcia, Juana Mari Navarro, “Functional Anatomy of thalamus and basal Ganlia”, *Child’s Nerv Syst* (2002) 18:386–404, DOI 10.1007/s00381-002-0604-1

[5] James Knierim, Ph.D., chapter 2 Basal Ganglia, Department of Neuroscience, The John Hopkins University

[6] Michael X. Cohen, Michael J. Frank, “ Neurocomputational models of basal ganglia function in learning, memory and choice”, *Department of Psychology, Program in Neuroscience*, University of Arizona, 1503 E University Blvd, Tucson, AZ 85721, United States

[7] Bradley Voytek, “Emergent Basal Ganglia Pathology within Computational Models”, *Helen Wills Neuroscience Institute, University of California, Berkeley,* California 94720-3190

[8] Constance Hammond, Hagai Bergman and Peter Brown, “Pathological synchronization in Parkinson’s disease: networks, models and Treatments”

[9] MICHAEL J. FRANK, BRYAN LOUGHRY, and RANDALL C. O'REILLY "Interactions between frontal cortex and basal ganglia in working memory: computational model" *Cognitive, Affective, & Behavioral Neuroscience* 2001, 1 (2), 137-160

[10] Frank, M.J. (2004), "Dynamic dopamine modulation of striato-cortical circuits in cognition: Converging neuropsychological, psychopharmacological and computational studies", *Phd Thesis, University of Colorado at Boulder, Boulder, CO*

[11] Frank, M.J. (2008), "Schizophrenia: A computational reinforcement learning perspective", *Schizophrenia Bulletin*, 34, 1008-1011

[12] Michael J. Frank_,"Hold your horses: A dynamic computational role for the subthalamic nucleus in decision making" *Neural Networks 19 (2006)* 1120–1136

[13] Thomas R (1973), "Boolean formalization of genetic control circuits", *J Theor Biol* 42: 563–585

[14] Thomas R, Gathoye AM, Lambert L (1976), "A complex control circuit: Regulation of immunity in temperate bacteriophages", *Eur J Biochem* 71: 211–227

[15] Thomas R (1979) , "Kinetic logic: a boolean approach to the analysis of complex regulatory systems", *Lecture Notes in Biomathematics* 29: 507

[16] Thomas R (1978), " Logical analysis of systems comprising feedback loops", *Journal of Theoretical Biology*, 73: 631–656

[17] Thomas R, "D'Ari R (1990) Biological Feedback", CRC *Press, Boca Raton*, FL

[18] Thomas R (1991), "Regulatory networks seen as asynchronous automata: A logical description", *Journal of Theoretical Biology* 153: 1–23

[19] Snoussi E, Thomas R (1993), "Logical identi_cation of all steady states : the concept of feedback loop characteristic states", *Bull Math Biol* 55: 973–991

[20] Thomas R, Thieffry D, Kaufman M (1995), "Dynamical behaviour of biological regulatory networks: I. biological role of feedback loops and practical use of the concept of the loop-characteristic state", *Bull Math Biol* 57: 247–276

[21] Thomas R (1998), "Laws for the dynamics of regulatory circuits" *Int J Dev Biol* 42:
479–485

[22] Ahmad J (2009) “Mode´lisation hybride et analyse des dynamiques des re´seaux de re´gulations biologiques en tenant compte des de´lais”, *Ph.D. thesis*, Ecole Centrale de Nantes, France.

[23] Gonzalez A, Naldi A, Sanchez L, Thieffry D, Chaouiya C (2006) “GINsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks”, *Biosystems* 84: 91–100

[24] Daniel Volk, “Population oscillations in a discrete model of neural networks of the brain” *BioSystems* 63 (2001) 35–41

[25] Ahmad J, Bernot G, Comet JP, Lime D, Roux O (2007), “Hybrid modelling and dynamical analysis of gene regulatory networks with delays”, *ComPlexUs 3*: 231–251

[26] Ade´lı¨ade M, Sutre G (2004) , “Parametric analysis and abstraction of genetic regulatory networks*”, In: Proc. 2nd Workshop on Concurrent Models in Molecular Biology (BioCONCUR’04), London, UK. Elsevier, Electronic Notes in Theor. Comp. Sci*.

[27] Ahmad J, Roux O, Bernot G, Comet J, Richard A (2008), “Analysing formal models of genetic regulatory networks with delays” *Int J Bioinformatics Res Appl 4: 240–262*

[28] Alexander Bockmayr, “Modeling biological systems in hybrid concurrent constraint programming”, Univ. Henri Poincare*”,´ LORIA B.P. 239, 54506 Vandœuvre, France*

[29] J. Ahmad and O. Roux, “Invariance kernel of biological regulatory networks”, Int. J. Data Min. Bioinformatics, vol4, pp.553-570, 2010

[30] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.H. Ho, “Hybrid automata: an algorithmic approach to the speci_cation and veri_cation of hybrid system”, *In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, Hybrid Systems I, Lecture Notes in Computer Science* 736, pages 209{229. Springer-Verlag, 1993

[31] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A.Olivero, J. Sifakis, and S. Yovine, “The algorithmic analysis of hybrid systems”, *Theoretical Computer Science,* 138:3{34, 1995

[32] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, “An approach to the description and analysis of hybrid systems”, *In R.L. Grossman, A. Nerode, A.P.*

Ravn, and H. Rischel, editors, Hybrid Systems I, Lecture Notes in Computer Science 736, pages 149 {178. Springer-Verlag, 1993

[33] Thomas A. Henzinger Pei-Hsin Ho Howard Wong-Toi, “A User Guide to HyTech”

[34] Thomas A. Henzinger1, Pei-Hsin Ho2, HowardWong-Toi, “HYTECH: a model checker for hybrid systems”, *Int J STTT (1997)* 1: 110{122 Ó 1997

[35] Koutsofios E, North SC (1993), “Drawing graphs with dot”

[36] Emden R. Gansner (2011) , “Drawing graphs with Graphviz”

[37] Jamil Ahmad, Umar Niazi, Sajid Mansoor, Umair Siddique,Jaclyn Bibbly (2012) “Formal Modeling and Analysis of the MAL-Associated Biological Regulatory Network: Insight into Cerebral Malaria” *PLoS ONE* 7(3): e33532 doi;10.1371/journal.pone0033532

[38] María-Trinidad Herrero, Carlos Barcia, Juana Mari Navarro “Functional anatomy of thalamus and basal ganglia”, Child’s Nerv Syst (2002) 18:386–404DOI 10.1007/s00381-002-0604-1

[39] José A. Obeso,María C. Rodríguez-Oroz,Manuel Rodríguez, Javier Arbizu, and José M. Giménez-Amaya, “The Basal Ganglia and Disorders of Movement: Pathophysiological Mechanisms”, *Departments of 1Neurology, 2Nuclear Medicine, and 3Anatomy, Clinica Universitaria and Medical School, University of Navarra, Pamplona 31008; and Department of Physiology, Medical School, University of La Laguna*, Tenerife 35042, Spain

[40] Helmich RC, Hallett M, Deuschl G, Toni I, Bloem BR, “Cerebral causes and consequences of parkinsonian resting tremors: A tale of two circuits. EPDA, November 2012

[41] Chaorui Huang, ChengkeTang, Andrew Feigin, Martin Lesser, Yilong Ma,1,2 Michael Pourfar,Vijay Dhawan1,2 and David Eidelberg, “Changes in network activity with the progression of Parkinson’s disease”, *Brain (2007)*, 130, 1834^1846

APPENDIX A

HyTech Code

```
var
hD2,hD1,hcor_thal,hGPI,hGPE,hSNC :analog;
dD2_u,dD1_u,dGPI_u,dGPE_u,dSNC_u, dcor_thal_u,
dD2_l,dD1_l,dGPI_l,dGPE_l,dSNC_l, dcor_thal_l: parameter;
automaton auto synclabs: ;
initially loc_000000;
-- gène n°0 = D2
-- gène n°1 = D1
-- gène n°2 = cor_thal
-- gène n°3 = GPI
-- gène n°4 = GPE
-- gène n°5 = SNC


----------------------------------------------------------------------------------------------------
--Define Values for respective variables
define(dGPI_u, 70)
define(dGPE_u, 20)


define(dcor_thal_u, 0)
define(dSNC_u, 50)

define(dD1_u, 10)

define(dD2_u, 20)

define(dGPI_l, 10)

define(dGPE_l, 10)

define(dcor_thal_l, 10)

define(dD1_l, 10)

define(dD2_l, 10)

----------------------------------------------------------------------------------------------------


-- pour la configuration 0,0,0,0,0,0

        loc loc_000000: while  hcor_thal <= dcor_thal_u & hGPI <= dGPI_u & hGPE

        <= dGPE_u & hSNC <= dSNC_u wait { dhD2=0, dhD1=0, dhcor_thal=1,

        dhGPI=1, dhGPE=1, dhSNC=1}

        --when hD2 = dD2_u do {hD2'=0} goto loc_100000;

        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001000;

        when hGPI = dGPI_u do {hGPI'=0} goto loc_000100;

        when hGPE = dGPE_u do {hGPE'=0} goto loc_000010;

        when hSNC = dSNC_u do {hSNC'=0} goto loc_000001;




-- pour la configuration 0,0,0,0,0,1
```

loc loc_000001: while hcor_thal <= dcor_thal_u & hGPI <= dGPI_u & hGPE <= dGPE_u wait {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=0}

--when hD1 = dD1_u do {hD1'=0} goto loc_010001;

when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001001;

when hGPI = dGPI_u do {hGPI'=0} goto loc_000101;

when hGPE = dGPE_u do {hGPE'=0} goto loc_000011;

-- *pour la configuration 0,0,0,0,1,0*

loc loc_000010: while hcor_thal <= dcor_thal_u & hSNC <= dSNC_u wait {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=1}

--when hD2 = dD2_u do {hD2'=0} goto loc_100010;

when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001010;

when hSNC = dSNC_u do {hSNC'=0} goto loc_000011;

-- *pour la configuration 0,0,0,0,1,1*

loc loc_000011: while hcor_thal <= dcor_thal_u wait { dhD2=0, dhD1=0, dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=0}

--when hD1 = dD1_u do {hD1'=0} goto loc_010011;

when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001011;

-- *pour la configuration 0,0,0,1,0,0*

```
        loc loc_000100: while  hGPE <= dGPE_u & hSNC <= dSNC_u wait
        {dhD2=0,dhD1=0,dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=1}
        --when hGPI = dGPI_l do {hGPI'=0} goto loc_000000;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_000110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_000101;

-- pour la configuration 0,0,0,1,0,1
        loc loc_000101: while hGPE <= dGPE_u wait { dhD2=0, dhD1=0,
        dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=0}
        --when hGPI = dGPI_l do {hGPI'=0} goto loc_000001;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_000111;

-- pour la configuration 0,0,0,1,1,0
        loc loc_000110: while hGPI <= dGPI_l & hSNC <= dSNC_u wait
        {dhD2=0,dhD1=0,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=1}
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001110;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_000010;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_000111;

-- pour la configuration 0,0,0,1,1,1
```

```
        loc loc_000111: while hGPI <= dGPI_l wait { dhD2=0, dhD1=0, dhcor_thal=1,
        dhGPI=0,dhGPE=0,dhSNC=0}
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_001111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_000011;

-- pour la configuration 0,0,1,0,0,0
        loc loc_001000: while hD2 <= dD2_u & hGPI <= dGPI_u & hGPE <= dGPE_u
        & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=0, dhGPI=1,
        dhGPE=1,dhSNC=1}
        when hD2 = dD2_u do {hD2'=0} goto loc_101000;
        --when hD1 = dD1_u do {hD1'=0} goto loc_011000;
        --when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000000;
        when hGPI = dGPI_u do {hGPI'=0}  goto loc_001100;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_001010;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_001001;

-- pour la configuration 0,0,1,0,0,1
        loc loc_001001: while hD1 <= dD1_u & hGPI <= dGPI_u & hGPE <= dGPE_u
        wait {dhD2=0,dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=1,dhSNC=0}
        when hD1 = dD1_u do {hD1'=0} goto loc_011001;
        --when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000001;


        when hGPI = dGPI_u do {hGPI'=0} goto loc_001101;
```

```
    when hGPE = dGPE_u do {hGPE'=0} goto loc_001011;


-- pour la configuration 0,0,1,0,1,0
    loc loc_001010: while  hD1 <= dD1_u & hSNC <= dSNC_u wait {dhD2=1,
    dhD1=0,dhcor_thal=0,dhGPI=0,dhGPE=0,dhSNC=1}
    when hD2 = dD2_u do {hD2'=0} goto loc_101010;
    --when hD1 = dD1_u do {hD2'=0} goto loc_011010;
    --when hGPE = dGPE_l do {hGPE'=0} goto loc_001000;
    when hSNC = dSNC_u do {hSNC'=0} goto loc_001011;


-- pour la configuration 0,0,1,0,1,1
    loc loc_001011: while hD1 <= dD1_u  wait {dhD2=0, dhD1=1, dhcor_thal=0,
    dhGPI=0,dhGPE=0,dhSNC=0}
    when hD1 = dD1_u do {hSNC'=0} goto loc_011011;


-- pour la configuration 0,0,1,1,0,0
    loc loc_001100: while hD2 <= dD2_u & hcor_thal <= dcor_thal_l & hGPE <=
    dGPE_u & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=1, dhGPI=0,
    dhGPE=1,dhSNC=1}
    when hD2 = dD2_u do {hD2'=0} goto loc_101100;
    --when hD1 = dD1_u do {hD1'=0} goto loc_011100;
    when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000100;
    --when hGPI = dGPI_l do {hGPI'=0} goto loc_001000;
```

```
        when hGPE = dGPE_u do {hGPE'=0} goto loc_001110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_001101;

-- pour la configuration 0,0,1,1,0,1
        loc loc_001101: while hD1 <= dD1_u & hcor_thal <= dcor_thal_l & hGPE <=
        dGPE_u  wait {dhD2=0,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=1,dhSNC=0}
        when hD1 = dD1_u do {hD1'=0} goto loc_011101;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000101;
        --when hGPI = dGPI_l do {hGPI'=0}  goto loc_001001;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_001111;

-- pour la configuration 0,0,1,1,1,0
        loc loc_001110: while hD2 <= dD2_u & hcor_thal <= dcor_thal_l & hGPI <=
        dGPI_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=1, dhGPI=1,
        dhGPE=0,dhSNC=1}
        when hD2 = dD2_u do {hD2'=0} goto loc_101110;
        --when hD1 = dD1_u do {hD1'=0} goto loc_011110;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000110;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_001010;
        --when hGPE = dGPE_l do {hGPE'=0} goto loc_001100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_001111;

-- pour la configuration 0,0,1,1,1,1
```

```
        loc loc_001111: while hD1 <= dD1_u & hcor_thal <= dcor_thal_l & hGPI <=
        dGPI_l wait {dhD2=0,dhD1=1,dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=0}
        when hD1 = dD1_u do {hD1'=0} goto loc_011111;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_000111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_001011;

-- pour la configuration 0,1,0,0,0,0
        loc loc_010000: while hD1 <= dD1_l & hcor_thal <= dcor_thal_u & hGPE <=
        dGPE_u & hSNC <= dSNC_u wait {dhD2=0, dhD1=1, dhcor_thal=1, dhGPI=0,
        dhGPE=1,dhSNC=1}
        --when hD2 = dD2_u do {hD2'=0} goto loc_110000;
        when hD1 = dD1_l do {hD1'=0} goto loc_000000;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011000;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_010100;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_010010;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_010001;

-- pour la configuration 0,1,0,0,0,1
        loc loc_010001: while hD1 <= dD1_l & hcor_thal <= dcor_thal_u & hGPE <=
        dGPE_u wait {dhD2=0,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=1,dhSNC=0}
        when hD1 = dD1_l do {hD1'=0} goto loc_000001;


        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011001;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_010101;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_010011;

-- pour la configuration 0,1,0,0,1,0
        loc loc_010010: while hD1 <= dD1_l & hcor_thal <= dcor_thal_u & hGPE <=
        dGPE_u wait {dhD2=0,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=1}
        --when hD2 = dD2_u do {hD2'=0} goto loc_110010;
        when hD1 = dD1_l do {hD1'=0} goto loc_000010;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011010;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_010011;

-- pour la configuration 0,1,0,0,1,1
        loc loc_010011: while hD1 <= dD1_l & hcor_thal <= dcor_thal_u wait
        {dhD2=0,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=0}
        when hD1 = dD1_l do {hD1'=0} goto loc_000011;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011011;

-- pour la configuration 0,1,0,1,0,0
        loc loc_010100: while hD1 <= dD1_l & hGPI <= dGPI_l & hGPE <= dGPE_u
        & hSNC <= dSNC_u wait {dhD2=0, dhD1=1, dhcor_thal=0, dhGPI=1,
        dhGPE=1,dhSNC=1}


        when hD1 = dD1_l do {hD1'=0} goto loc_000100;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011100;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_010000;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_010110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_010101;

-- pour la configuration 0,1,0,1,0,1
        loc loc_010101: while hD1 <= dD1_l & hGPI <= dGPI_l & hGPE <= dGPE_u
        wait {dhD2=0,dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=1,dhSNC=0}
        when hD1 = dD1_l do {hD1'=0} goto loc_000101;
        --when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_011101;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_010001;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_010111;

-- pour la configuration 0,1,0,1,1,0
        loc loc_010110: while hD1 <= dD1_l & hGPI <= dGPI_l & hSNC <= dSNC_u
        wait {dhD2=0,dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=1}
        when hD1 = dD1_l do {hD1'=0} goto loc_000110;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011110;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_010010;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_010111;


-- pour la configuration 0,1,0,1,1,1
```

```
        loc loc_010111: while hD1 <= dD1_l & hGPI <= dGPI_l wait {dhD2=0,
        dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=0}
        when hD1 = dD1_l do {hD1'=0} goto loc_000111;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_011111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_010011;

-- pour la configuration 0,1,1,0,0,0
        loc loc_011000: while hD2 <= dD2_u & hD1 <= dD1_l & hGPE <= dGPE_u &
        hSNC <= dSNC_u wait {dhD2=1, dhD1=1, dhcor_thal=0, dhGPI=0, dhGPE=1,
        dhSNC=1}
        when hD2 = dD2_u do {hD2'=0} goto loc_111000;
        when hD1 = dD1_l do {hD1'=0} goto loc_001000;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_011100;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_011010;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_011001;

-- pour la configuration 0,1,1,0,0,1
        loc loc_011001: while hGPE = dGPE_u wait {dhD2=0, dhD1=0, dhcor_thal=0,
        dhGPI=0,dhGPE=1,dhSNC=0}
        when hGPE = dGPE_u do {hGPE'=0} goto loc_011011;



-- pour la configuration 0,1,1,0,1,0
```

```
        loc loc_011010: while hD2 <= dD2_u & hD1 <= dD1_l & hSNC <= dSNC_u
        wait {dhD2=1,dhD1=1,dhcor_thal=0,dhGPI=0,dhGPE=0,dhSNC=1}
        when hD2 = dD2_u do {hD2'=0} goto loc_111010;
        when hD1 = dD1_l do {hD1'=0} goto loc_001010;
        --when hGPE = dGPE_l do {hGPE'=0} goto loc_011000;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_011011;

-- pour la configuration 0,1,1,0,1,1
        loc loc_011011: while True wait {dhD2=0, dhD1=0, dhcor_thal=0, dhGPI=0,
        dhGPE=0,dhSNC=0}

-- pour la configuration 0,1,1,1,0,0
        loc loc_011100: while hD2 <= dD2_u & hD1 <= dD1_l & hcor_thal <=
        dcor_thal_l & hGPI <= dGPI_l & hGPE <= dGPE_u & hSNC <= dSNC_u wait
        {dhD2=1,dhD1=1,dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=1}
        when hD2 = dD2_u do {hD2'=0} goto loc_111100;
        when hD1 = dD1_l do {hD1'=0} goto loc_001100;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_010100;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_011000;
        when hGPE = dGPE_u do {hGPE'=0} goto loc_011110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_011101;

-- pour la configuration 0,1,1,1,0,1
```

loc loc_011101: while hcor_thal <= dcor_thal_l & hGPI <= dGPI_l & hGPE <= dGPE_u wait {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=0}

--when hD1 = dD1_l do {hD1'=0} goto loc_001101;

when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_010101;

when hGPI = dGPI_l do {hGPI'=0} goto loc_011001;

when hGPE = dGPE_u do {hGPE'=0} goto loc_011111;

-- *pour la configuration 0,1,1,1,1,0*

loc loc_011110: while hD2 <= dD2_u & hD1 <= dD1_l & hcor_thal <= dcor_thal_l & hGPI <= dGPI_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=1,dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=1}

when hD2 = dD2_u do {hD2'=0} goto loc_111110;

when hD1 = dD1_l do {hD1'=0} goto loc_001110;

when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_010110;

when hGPI = dGPI_l do {hGPI'=0} goto loc_011010;

--when hGPE = dGPE_l do {hGPE'=0} goto loc_011100;

when hSNC = dSNC_u do {hSNC'=0} goto loc_011111;

-- *pour la configuration 0,1,1,1,1,1*

loc loc_011111: while hcor_thal <= dcor_thal_l & hGPI <= dGPI_l wait {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=0}

--when hD1 = dD1_l do {hD1'=0} goto loc_001111;

```
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_010111;

        when hGPI = dGPI_l do {hGPI'=0} goto loc_011011;


-- pour la configuration 1,0,0,0,0,0

        loc loc_100000: while hD2 <= dD2_l & hcor_thal <= dcor_thal_u & hGPI <=
        dGPI_u & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=1, dhGPI=1,
        dhGPE=0,dhSNC=1}

        when hD2 = dD2_l do {hD2'=0} goto loc_000000;

        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101000;

        when hGPI = dGPI_u do {hGPI'=0} goto loc_100100;

        --when hGPE = dGPE_u do {hGPE'=0} goto loc_100010;

        when hSNC = dSNC_u do {hSNC'=0} goto loc_100001;


-- pour la configuration 1,0,0,0,0,1

        loc loc_100001: while hD2 <= dD2_l & hcor_thal <= dcor_thal_u & hGPI <=
        dGPI_u wait {dhD2=1,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=0}

        when hD2 = dD2_l do {hD2'=0} goto loc_000001;

        --when hD1 = dD1_u do {hD1'=0} goto loc_110001;

        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101001;

        when hGPI = dGPI_u do {hGPI'=0} goto loc_100101;

        --when hGPE = dGPE_u do {hGPE'=0} goto loc_100011;


-- pour la configuration 1,0,0,0,1,0
```

```
        loc loc_100010: while hD2 <= dD2_l & hcor_thal <= dcor_thal_u & hGPE <=
        dGPE_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=1, dhGPI=0,
        dhGPE=1,dhSNC=1}
        when hD2 = dD2_l do {hD2'=0} goto loc_000010;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101010;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_100110;
        when hGPE =   dGPE_l do {hGPE'=0} goto loc_100000;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_100011;


-- pour la configuration 1,0,0,0,1,1
        loc loc_100011: while hD2 <= dD2_l & hD1 <= dD1_u & hcor_thal <=
        dcor_thal_u & hGPE <=  dGPE_l wait {dhD2=1, dhD1=1, dhcor_thal=1,
        dhGPI=0,dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_000011;
        when hD1 = dD1_u do {hD1'=0} goto loc_110011;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101011;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_100111;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_100001;




-- pour la configuration 1,0,0,1,0,0
```

```
        loc loc_100100: while hD2 <= dD2_l & hSNC <= dSNC_u wait {dhD2=1,
        dhD1=0,dhcor_thal=0,dhGPI=0,dhGPE=0,dhSNC=1}
        when hD2 = dD2_l do {hD2'=0} goto loc_000100;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_100110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_100101;

-- pour la configuration 1,0,0,1,0,1
        loc loc_100101: while hD2 <= dD2_l wait {dhD2=1, dhD1=0, dhcor_thal=0,
        dhGPI=0,dhGPE=0,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_000101;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_100111;

-- pour la configuration 1,0,0,1,1,0
        loc loc_100110: while hD2 <= dD2_l & hGPI <= dGPI_l & hGPE <=  dGPE_l
        & hSNC <= dSNC_u wait {dhD2=1, dhD1=0, dhcor_thal=0, dhGPI=1,
        dhGPE=1, dhSNC=1}
        when hD2 = dD2_l do {hD2'=0} goto loc_000110;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101110;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_100010;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_100100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_100111;

-- pour la configuration 1,0,0,1,1,1
```

```
        loc loc_100111: while hD2 <= dD2_l & hGPI <= dGPI_l & hGPE <=  dGPE_l
        wait {dhD2=1,dhD1=0,dhcor_thal=0,dhGPI=1,dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_000111;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_101111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_100011;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_100101;

-- pour la configuration 1,0,1,0,0,0
        loc loc_101000: while hGPI <= dGPI_u & hSNC <= dSNC_u wait {dhD2=0,
        dhD1=0,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_001000;
        --when hD1 = dD1_u do {hD1'=0} goto loc_111000;
        --when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100000;
        when hGPI = dGPI_u do {hGPI'=0} goto loc_101100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_101001;

-- pour la configuration 1,0,1,0,0,1
        loc loc_101001: while hD2 <= dD2_l & hD1 <= dD1_u & hGPI <= dGPI_u wait
        {dhD2=1,dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_001001;
        when hD1 = dD1_u do {hD1'=0} goto loc_111001;


        --when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100001;
        when hGPI = dGPI_u do {hGPI'=0} goto loc_101101;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_101011;


-- pour la configuration 1,0,1,0,1,0
        loc loc_101010: while  hGPE <=  dGPE_l & hSNC <= dSNC_u wait {dhD2=0,
        dhD1=0,dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_001010;
        --when hD1 = dD1_u do {hD1'=0} goto loc_111010;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_101110;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_101000;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_101011;


-- pour la configuration 1,0,1,0,1,1
        loc loc_101011: while hD2 <= dD2_l & hD1 <= dD1_u & hGPE <=  dGPE_l
        wait {dhD2=1,dhD1=1,dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_001011;
        when hD1 = dD1_u do {hD1'=0} goto loc_111011;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_101111;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_101001;




-- pour la configuration 1,0,1,1,0,0
```

```
        loc loc_101100: while hcor_thal <= dcor_thal_l & hSNC <= dSNC_u wait
        {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_001100;
        --when hD1 = dD1_u do {hD1'=0} goto loc_111100;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_101101;

-- pour la configuration 1,0,1,1,0,1
        loc loc_101101: while hD2 <= dD2_l & hcor_thal <= dcor_thal_l & hD1 <=
        dD1_u wait {dhD2=1,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_001101;
        when hD1 = dD1_u do {hD1'=0} goto loc_111101;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100101;
        --when hGPI = dGPI_l do {hGPI'=0} goto loc_101001;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_101111;

-- pour la configuration 1,0,1,1,1,0
        loc loc_101110: while hcor_thal <= dcor_thal_l & hGPI <= dGPI_l & hGPE <=
        dGPE_l & hSNC <= dSNC_u wait
        {dhD2=0,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_001110;


        --when hD1 = dD1_u do {hD1'=0} goto loc_111110;
```

```
       when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100110;
       when hGPI = dGPI_l do {hGPI'=0} goto loc_101010;
       when hGPE =  dGPE_l do {hGPE'=0} goto loc_101100;
       when hSNC = dSNC_u do {hSNC'=0} goto loc_101111;

-- pour la configuration 1,0,1,1,1,1
       loc loc_101111: while hD2 <= dD2_l & hD1 <= dD1_u & hcor_thal <=
       dcor_thal_l & hGPI <= dGPI_l & hGPE <=  dGPE_l wait {dhD2=1, dhD1=1,
       dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=0}
       when hD2 = dD2_l do {hD2'=0} goto loc_001111;
       when hD1 = dD1_u do {hD1'=0} goto loc_111111;
       when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_100111;
       when hGPI = dGPI_l do {hGPI'=0} goto loc_101011;
       when hGPE =  dGPE_l do {hGPE'=0} goto loc_101101;

-- pour la configuration 1,1,0,0,0,0
       loc loc_110000: while hD2 <= dD2_l & hD1 <= dD1_l & hcor_thal <=
       dcor_thal_u & hSNC <= dSNC_u wait
       {dhD2=1,dhD1=1,dhcor_thal=1,dhGPI=0,dhGPE=0,dhSNC=1}
       when hD2 = dD2_l do {hD2'=0} goto loc_010000;
       when hD1 = dD1_l do {hD1'=0} goto loc_100000;
       when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111000;
       --when hGPI = dGPI_u do {hGPI'=0} goto loc_110100;
```

```
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_110010;

        when hSNC = dSNC_u do {hSNC'=0} goto loc_110001;


-- pour la configuration 1,1,0,0,0,1

        loc loc_110001: while hD2 <= dD2_l & hD1 <= dD1_l & hcor_thal <=
        dcor_thal_u wait {dhD2=1, dhD1=1, dhcor_thal=1, dhGPI=0, dhGPE=0,
        dhSNC=0}

        when hD2 = dD2_l do {hD2'=0} goto loc_010001;

        when hD1 = dD1_l do {hD1'=0} goto loc_100001;

        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111001;

        --when hGPI = dGPI_u do {hGPI'=0} goto loc_110101;

        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_110011;


-- pour la configuration 1,1,0,0,1,0

        loc loc_110010: while hD2 <= dD2_l & hD1 <= dD1_l & hcor_thal <=
        dcor_thal_u & hGPE <=  dGPE_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=1,
        dhcor_thal=1,dhGPI=0,dhGPE=1,dhSNC=1}

        when hD2 = dD2_l do {hD2'=0} goto loc_010010;

        when hD1 = dD1_l do {hD1'=0} goto loc_100010;

        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111010;

        when hGPE =  dGPE_l do {hGPE'=0} goto loc_110000;


        when hSNC = dSNC_u do {hSNC'=0} goto loc_110011;
```

```
-- pour la configuration 1,1,0,0,1,1
        loc loc_110011: while hD2 <= dD2_l & hD1 <= dD1_l & hcor_thal <=
        dcor_thal_u & hGPE <=  dGPE_l wait {dhD2=1, dhD1=1, dhcor_thal=1,
        dhGPI=0,dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_010011;
        when hD1 = dD1_l do {hD1'=0} goto loc_100011;
        when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111011;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_110001;

-- pour la configuration 1,1,0,1,0,0
        loc loc_110100: while hD2 <= dD2_l & hD1 <= dD1_l & hGPI <= dGPI_l &
        hSNC <= dSNC_u wait {dhD2=1, dhD1=1, dhcor_thal=0, dhGPI=1, dhGPE=0,
        dhSNC=1}
        when hD2 = dD2_l do {hD2'=0} goto loc_010100;
        when hD1 = dD1_l do {hD1'=0} goto loc_100100;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111100;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_110000;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_110110;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_110101;


-- pour la configuration 1,1,0,1,0,1
```

```
        loc loc_110101: while hD2 <= dD2_l & hD1 <= dD1_l & hGPI <= dGPI_l wait
        {dhD2=1,dhD1=1,dhcor_thal=0,dhGPI=1,dhGPE=0,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_010101;
        when hD1 = dD1_l do {hD1'=0} goto loc_100101;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111101;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_110001;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_110111;

-- pour la configuration 1,1,0,1,1,0
        loc loc_110110: while hD2 <= dD2_l & hD1 <= dD1_l & hGPI <= dGPI_l &
        hGPE <=  dGPE_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=1, dhcor_thal=0,
        dhGPI=1,dhGPE=1,dhSNC=1}
        when hD2 = dD2_l do {hD2'=0} goto loc_010110;
        when hD1 = dD1_l do {hD1'=0} goto loc_100110;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111110;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_110010;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_110100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_110111;

-- pour la configuration 1,1,0,1,1,1
```

```
        loc loc_110111: while hD2 <= dD2_l & hD1 <= dD1_l & hGPI <= dGPI_l &
        hGPE <=  dGPE_l wait {dhD2=1, dhD1=1, dhcor_thal=0, dhGPI=1, dhGPE=1,
        dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_010111;
        when hD1 = dD1_l do {hD1'=0} goto loc_100111;
        --when hcor_thal = dcor_thal_u do {hcor_thal'=0} goto loc_111111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_110011;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_110101;

-- pour la configuration 1,1,1,0,0,0
        loc loc_111000: while hD1 <= dD1_l & hSNC <= dSNC_u wait {dhD2=0,
        dhD1=1,dhcor_thal=0,dhGPI=0,dhGPE=0,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_011000;
        when hD1 = dD1_l do {hD1'=0} goto loc_101000;
        --when hGPI = dGPI_u do {hGPI'=0} goto loc_111100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_111001;

-- pour la configuration 1,1,1,0,0,1
        loc loc_111001: while  hD2 = dD2_l wait {dhD2=1, dhD1=0, dhcor_thal=0,
        dhGPI=0, dhGPE=0,dhSNC=0}



        when hD2 = dD2_l do {hD2'=0} goto loc_011001;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_111011;


-- pour la configuration 1,1,1,0,1,0
        loc loc_111010: while hD1 <= dD1_l & hGPE <=  dGPE_l & hSNC <= dSNC_u
        wait {dhD2=0,dhD1=1,dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_011010;
        when hD1 = dD1_l do {hD1'=0} goto loc_101010;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_111000;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_111011;


-- pour la configuration 1,1,1,0,1,1
        loc loc_111011: while hD2 <= dD2_l & hGPE <=  dGPE_l wait {dhD2=1,
        dhD1=0,dhcor_thal=0,dhGPI=0,dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_011011;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_111001;


-- pour la configuration 1,1,1,1,0,0
        loc loc_111100: while hD2 <= dD2_l & hD1 <= dD1_l & hcor_thal <=
        dcor_thal_l & hGPI <= dGPI_l & hSNC <= dSNC_u wait {dhD2=1, dhD1=1,
        dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=1}



        when hD2 = dD2_l do {hD2'=0} goto loc_011100;
        when hD1 = dD1_l do {hD1'=0} goto loc_101100;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0}  goto loc_110100;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_111000;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_111101;

-- pour la configuration 1,1,1,1,0,1
        loc loc_111101: while hD2 <= dD2_l & hcor_thal <= dcor_thal_l & hGPI <=
        dGPI_l wait {dhD2=1,dhD1=0,dhcor_thal=1,dhGPI=1,dhGPE=0,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_011101;
        --when hD1 = dD1_l do {hD1'=0} goto loc_101101;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_110101;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_111001;
        --when hGPE =  dGPE_u do {hGPE'=0} goto loc_111111;

-- pour la configuration 1,1,1,1,1,0
        loc loc_111110: while hD1 <= dD1_l & hcor_thal <= dcor_thal_l & hGPI <=
        dGPI_l & hGPE <=  dGPE_l & hSNC <= dSNC_u wait {dhD2=0, dhD1=1,
        dhcor_thal=1,dhGPI=1,dhGPE=1,dhSNC=1}
        --when hD2 = dD2_l do {hD2'=0} goto loc_011110;
        when hD1 = dD1_l do {hD1'=0} goto loc_101110;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_110110;


        when hGPI = dGPI_l do {hGPI'=0} goto loc_111010;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_111100;
        when hSNC = dSNC_u do {hSNC'=0} goto loc_111111;


-- pour la configuration 1,1,1,1,1,1
        loc loc_111111: while hD2 <= dD2_l & hcor_thal <= dcor_thal_l & hGPI <=
        dGPI_l & hGPE <=  dGPE_l wait {dhD2=1, dhD1=0, dhcor_thal=1, dhGPI=1,
        dhGPE=1,dhSNC=0}
        when hD2 = dD2_l do {hD2'=0} goto loc_011111;
        --when hD1 = dD1_l do {hD1'=0} goto loc_101111;
        when hcor_thal = dcor_thal_l do {hcor_thal'=0} goto loc_110111;
        when hGPI = dGPI_l do {hGPI'=0} goto loc_111011;
        when hGPE =  dGPE_l do {hGPE'=0} goto loc_111101;


end



-- The operators "&" and "~" represent the conjunction and negation
--operations respectively. --/Region declarations
var
ini_reg, fin_reg,acces: region;
--/ To find the path between two states first modify the


--  variables first_state, final_state, ir and fr
--  initial region

ini_reg:= loc[auto] = loc_000000 & hcor_thal=0 & hD1=0 & hD2=0 & hSNC=0 &

hGPI=0 & hGPE=0;

-- final region

fin_reg:= loc[auto] = loc_000000;

-- access all reachable states

acces:=post(post(post(post(post(post(post(post(ini_reg))))))));

--ini_reg:= hull(acces) & ini_reg;

--ini_reg:= hull(acces) & ini_reg;


prints "=====================================================";

prints "Accessible states from the initial state along with the

parameter constraints"; print hide non_parameters in acces endhide;

prints "=====================================================";
```