JOÃO HENRIQUE RANHEL RIBEIRO

COMPUTAÇÃO POR ASSEMBLEIAS NEURAS EM
REDES NEURAS PULSADAS

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências.

São Paulo
2012

JOÃO HENRIQUE RANHEL RIBEIRO

COMPUTAÇÃO POR ASSEMBLEIAS NEURAIIS EM
REDES NEURAIIS PULSADAS

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências.

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Marcio Lobo Netto

São Paulo
2012

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 16 de janeiro de 2012.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Ribeiro, João Henrique Ranhel

**Computação por assembléias neurais em redes neurais pul-
sadas / J.H.R. Ribeiro. -- ed.rev. -- São Paulo, 2012.
220 p.**

**Tese (Doutorado) - Escola Politécnica da Universidade de
São Paulo. Departamento de Engenharia de Sistemas Eletrô-
nicos.**

**1. Computação bioinspirada 2. Sistemas dinâmicos 3. Coali-
zões neurais 4. Grupos neurais policronizados 5. Assembléias
biestáveis I. Universidade de São Paulo. Escola Politécnica. De-
partamento de Engenharia de Sistemas Eletrônicos II. t.**

JOÃO RANHEL

COMPUTAÇÃO POR ASSEMBLEIAS NEURAIIS EM REDES NEURAIIS PULSADAS

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências.

Aprovado em: __05 / 12 / 2011__

Banca Examinadora:

Prof. Dr. __*Marcio Lobo Netto*__

Instituição __ PSI / Escola Politécnica da USP__

Prof. Dr. __*Gilberto Fernando Xavier*__

Instituição __ Instituto de Biociências - USP__

Prof. Dr. __*Ricardo Ribeiro Gudwin*__

Instituição __ DCA / FEEC / UNICAMP__

Prof. Dr. __*Luiz Antonio Baccalá*__

Instituição __ PTC / Escola Politécnica da USP__

Prof. Dr. __*Emilio Del Moral Hernandez*__

Instituição __ PSI / Escola Politécnica da USP__

DEDICATÓRIA

À minha esposa, Cacilda, pelo amor e companheirismo, pelo apoio de várias formas, e principalmente, por ter ouvido e discutido tantos assuntos que, sorrateiros, se incorporaram ao seu cotidiano durante estes anos.

AGRADECIMENTOS

Agradeço a todos os meus familiares, meus pais, minha esposa, minha sogra, irmãos; afinal, tem um pedacinho de cada um neste trabalho.

Agradeço ao meu orientador, Prof. Dr. Marcio Lobo Netto por ter me propiciado a chance de realizar um velho sonho: estudar na Escola Politécnica da USP.

Agradeço ao saudoso Dr. Henrique Del Nero por ter-me “acicatado” em nossas proveitosas discussões.

Agradeço ainda aos colegas com os quais interagi na USP, em especial aos amigos do Grupo Cognitio: Ana Contier, Antonio C. Godoi, Cesar Miguel, Daniel Paiva, Danilo Bellini, Fabius Leineweber, Flávio Yamamoto, Frank Cara, Gustavo Patiño, João Kogler Jr., Júlio Monteiro, Luciene Rinaldi, Mariana Aguiar, Marcos Cavallieri, Mauro Muñoz, Osvaldo Guimarães, Reginaldo I. Filho, Walter Lima, dentre outros. Obrigado pelas inúmeras e frutíferas trocas de ideias.

Mário Quintana tem uma síntese na forma de poesia que diz:

*“Todos estes que aí estão
Atravancando o meu caminho.
Eles passarão...
Eu passarinho!”*

Agradeço a todos aqueles que atravancaram o meu caminho. Eles me forçaram a ver o mundo do ângulo pelo qual o enxergam os passarinhos.

"I believe in intuition and inspiration. Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution. It is, strictly speaking, a real factor in scientific research."

Albert Einstein

RESUMO

Um dos grandes mistérios da ciência é compreender como sistemas nervosos são capazes de realizar as extraordinárias operações computacionais que realizam. Provavelmente, encéfalos são as estruturas nas quais energia e matéria estão organizadas da forma mais complexa no universo. Central na computação cerebral está o conceito de neurônio. A forma como neurônios computam é motivo de intensa investigação científica. Um consenso atual é que neurônios formam grupos transientes (assembleias) a fim de representar coisas, de realizar operações computacionais, e de executar processos cognitivos; embora os mecanismos que fundamentam a “computação por assembleias” ainda não seja bem compreendido. Aqui é proposta uma *forma* pela qual se explica *como* computação por assembleias pode acontecer. Dois componentes são fundamentais para formação de coalizões neurais: a relação temporal entre grupos de neurônios e o fator de acoplamento entre eles. “Assembleias” pressupõe neurônios pulsantes; portanto, simulamos computação por assembleias em redes neurais pulsantes. A abordagem usada nesta tese é funcional; apresentamos um arcabouço teórico sobre propriedades, princípios, e dinâmicas que permitem operações computacionais por coalizões neurais. É apresentado na tese que: (i) quando neurônios formam assembleias está implícito que um tipo de função lógica estocástica ocorre, (ii) assembleias podem formar grupos com feedback, criando grupos biestáveis, (iii) grupos biestáveis criam representações internas dos eventos que os criaram, (iv) assembleias podem se ramificar e também dissolver outras assembleias, o que dá origem a algoritmos complexos. Esta é uma investigação inicial sobre computação em assembleias neurais, e há muito a ser feito. Nesta tese apresentamos os conceitos basais para esta nova abordagem. Há um conjunto de programas nos apêndices que permitem ao leitor simular formações de assembleias, ramificações, inibições, reverberações, entre outras propriedades e componentes de nossa proposta.

Palavras-chave: Assembleias Neurais. Coalizões Neurais. Assembleias Biestáveis. Grupos Policronizados. Cadeias Neurais Sincronizadas. Neurocomputação. Redes Neurais Pulsadas. Computação Bioinspirada. Neurociência.

ABSTRACT

One of the greatest mysteries in science is to comprehend how brains are capable of realizing the extraordinary computational operations they do. Probably, brains are the structures in which matter and energy are organized in the most complex way in the Universe. Central to the brain computation is the concept of neuron. How neurons compute is motive of intensive scientific investigation. A prevailing consensus is that neurons form transient groups (assemblies) in order to represent things, for realizing computational operations, and for executing cognitive processes; although the mechanisms that substantiate such “computation by neural assemblies” are not yet well understood. In this thesis we propose a *form* that explains *how* neural assembly computation may occur. It is shown that two components are fundamentals for neural coalition formation: the temporal relation among neural groups, and the coupling factor among them. In this sense, neural assemblies presuppose spiking neurons; therefore, here we simulate assembly computing using spiking neural networks. In this thesis it is presented basically a functional approach; thus, it presents a theoretical approach concerning the properties, principles, characteristics, and components that allow the computational operations in neural coalitions. It is presented in the thesis that: (i) as neurons form assemblies it is implicit that a kind of stochastic logic function occurs; (ii) assemblies may form groups that feedback each other, creating bistable groups; (iii) bistable groups internally represent the event that created them; (iv) assemblies may branch and dissolve other assemblies, what give rise to complex algorithms. This is an initial investigation about neural assembly computing and there is a lot to be done; however, in this thesis we present the basal concepts for this new approach. There are programs in the appendices that allow the reader to simulate assembly formation, branching, inhibition, reverberation, among other properties and components in our proposal.

Keywords: Neural Assembly. Neural Coalition. Bistable Assemblies. Polychronous Groups. Synfire Chains. Neurocomputing. Bioinspired Computing. Spiking Neural Networks. Neuroscience.

LISTA DE ABREVIATURAS E SIGLAS

aEIF	<i>adaptive exponential integrate-and-fire</i> (modelo neural)
BCD	binary decimal coded (codificação decimal em binário)
BCM	Bienenstock, Cooper e Munro (regra de plasticidade)
Ca ⁺⁺	íons de cálcio
CPG	<i>central pattern generator</i> (gerador central de padrões)
EPUSP	Escola Politécnica da Universidade de São Paulo
FLOP	<i>floating-point operation</i> (operação com ponto flutuante)
GP	grupo policronizado
GPB	grupo policronizado biestável (assembleia biestável)
GPU	graphic processor unity
H&H	Hodgkin e Huxley (modelo neural)
I&F	integrate and fire (modelo neural)
LIF	leaky integrate-and-fire (modelo neural)
LTD	<i>long-term depression</i> (depressão sináptica de longo prazo)
LTP	<i>long-term potentiation</i> (potenciação sináptica de longo prazo)
M&P	McCulloch e Pitts (modelo neural)
NA	neurônio artificial
Na ⁺	íons de sódio
K ⁺	íons de potássio
PA	potencial de ação, potenciais de ação (<i>spikes</i>)
PCA	<i>principal component analysis</i>
PCO	<i>pulse-coupled oscillators</i>
PEPS	potencial excitatório pós-sináptico
PH	plasticidade homeostática
PIPS	potencial inibitório pós-sináptico
PLL	<i>phase-locked loop</i>
QIF	<i>quadratic integrate-and-fire</i>
R-C	circuito elétrico formado por resistor e capacitor
RNA	redes neurais artificiais
RNR	redes neurais recorrentes
RNP	redes neurais pulsadas
RNPA	rede neural de pulso acoplado
SN(s)	sistema nervoso (ou sistemas nervosos)
SRM	<i>spike response model</i>
STDP	<i>spike-timing dependent plasticity</i>

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

1 INTRODUÇÃO	13
1.1 Objetivos	18
1.2 Computação não-padronizada e algoritmo	19
1.3 Motivação e justificativa	20
1.4 Revisão da literatura.....	21
1.5 Publicações associadas a esta tese	23
1.6 Organização da tese	24
1.7 Hipóteses, premissas e conjecturas que guiam a tese	25
2 COMPUTAÇÃO NEURAL	29
2.1 Níveis de investigação cerebrais	29
2.2 Neurônios e neuroides	33
2.3 Modelando neurônios	35
2.4 Breve histórico da simulação neuronal	37
2.5 O que há na temporização dos pulsos?	42
2.6 Reprodução do comportamento neural	47
2.6.1 Modelo de Hodgkin–Huxley	48
2.7 Modelos de neuroides	50
2.7.1 Modelo integra-e-dispara (I&F)	51
2.7.2 Família I&F	53
2.7.3 Modelo simplificado de Izhikevich	55
2.7.4 Modelo de resposta impulsiva	58

2.8 Resumo do capítulo	59
3 PLASTICIDADES	63
3.1 Plasticidade celular	63
3.2 Plasticidades de curto-prazo (<i>short-term plasticity</i>)	67
3.3 Plasticidade hebbiana (<i>long-term plasticity</i>)	70
3.4 Plasticidade homeostática	73
3.5 Unindo as regras de plasticidade	78
3.6 Plasticidade estrutural: reconfigurando conexões	81
3.7 Computação dendrítica	85
3.7.1 Aprendizado temporal	91
3.8 Resumo do capítulo	99
4 REDES E ASSEMBLEIAS	104
4.1 Redes neurais pulsadas recorrentes	105
4.1.1 Estratégia para simulação de RNPs	107
4.2 Marca-passo (<i>pacemakers</i>) e osciladores	108
4.2.1 Redes de osciladores e CPGs	110
4.3 Atrasos e frequências	113
4.4 Formação de assembleias	117
4.4.1 Grupos síncronos e policronizados (GP)	119
4.4.2 Assembleias biestáveis (GPB)	123
4.5 Resumo do capítulo	125
5 ALGORITMOS EM ASSEMBLEIAS NEURAIS.....	129
5.1 Ramificações e dissoluções de assembleias	131
5.2 Elementos da computação por assembleias	137
5.2.1 Funções lógicas estatísticas	137

5.2.2 Hierarquias e relações causais	139
5.3 Semântica e comportamento do exemplo	141
5.3.1 Representação interna	143
5.4 Questões sobre computação por assembleias	144
5.4.1 Assembleias multiestáveis	145
5.4.2 Recorrência dos eventos de disparo	150
5.4.3 Deslocamento de fase	154
5.5 Resumo do capítulo	157
6 SIMULAÇÕES E RESULTADOS.....	160
6.1 Ambiente de simulação	161
6.2 Transdução de sinais	163
6.3 Contadores digitais binários e acumuladores	169
6.4 Contadores com retardo e rede de pulso acoplado	174
6.5 Simulação de base para assembleias	177
6.6 Resumo do capítulo	188
7 CONCLUSÕES	190
7.1 Dos objetivos e resultados	194
7.2 Perspectivas e continuidade do trabalho	194
REFERÊNCIAS.....	197
APÊNDICE A	209
APÊNDICE B	211
APÊNDICE C	213
APÊNDICE D	216
APÊNDICE E	219

1 INTRODUÇÃO

Sistemas nervosos (SN) adaptam respostas e modificam comportamentos em ambientes sob constantes modificações. Sistemas nervosos encontram meios para trocar informações com outros SNs. Eles criam linguagens, elaboram pensamentos, constroem raciocínios. SNs descobrem leis que regem o universo em que vivem. Há que se ter admiração e respeito pela forma como a natureza e organismos estruturaram matéria e energia numa área tão diminuta, consumindo tão pouca energia, e com capacidades computacionais assombrosas. É preciso observar e revisitar cada detalhe dos sistemas nervosos dos organismos para tentar compreender esse conjunto estupendo de fatos que são o surgimento, a evolução e a existência de sistemas nervosos.

É provável que um dia essas caixas diminutas sejam capazes de explicar leis que descrevam seus próprios funcionamentos. Esta tese é sobre esse assunto. É sobre como cérebros (SNs) podem tirar proveito de ínfimos intervalos de tempos e coincidências entre pulsos, tão estreitos quanto um milésimo de segundo, para executarem computação que resulta em controlar o toque de dedos sobre um teclado, que cria uma série de manchas num fundo branco, que faz reverberar sons na cabeça de quem os lê, que transmite estados mentais, conceitos, raciocínios. Assim ocorre com uma nova noção aqui proposta, sobre como possivelmente funciona a computação por assembleias em redes de neurônios pulsantes.

Grande parte dos trabalhos científicos começa com um propósito, uma meta, apontando para uma direção, e no meio da jornada investigativa encontra bifurcações, desvios; não foi diferente com esta pesquisa. A intenção inicial era descobrir um conjunto de atributos computacionais a fim de orientar projetos de agentes e robôs *cognitivos*, além de compreender como certo tipo de computação acontece em redes neurais, tentando imitar, na medida do possível, o que ocorre em organismos. O primeiro grande problema, portanto, seria compreender o que é cognição. Como seria possível criar máquinas cognitivas se não soubermos o que colocar nelas? Quais seriam os parâmetros para justificar que um agente se comporta de forma cognitiva?

Entender o que é *cognição*, como ela evoluiu e como acontece é deveras importante porque, em primeiro lugar, é conhecer melhor uma parte de como somos, como agimos e reagimos em diversos contextos. Conhecer como os processos cognitivos acontecem pode abrir várias perspectivas em Neurociência, Biologia, Ciência Cognitiva, Computação e Matemática; e ainda para as Engenharias, que poderão criar máquinas que melhor interajam com humanos. Assim, a trilha primeira para nossa pesquisa seria identificar o que podemos chamar de “minimamente cognitivo”.

Porém, essa é uma tarefa hercúlea, com desdobramentos em Biologia, Filosofia, Paleontologia, Neurociência, e outras disciplinas. A pesquisa sobre cognição mínima serviu como guia até chegar a um ponto em que era necessário explicar como operações minimamente cognitivas acontecem em neurônios; nesse momento, foi preciso restringir a pesquisa.

Na primeira parte do livro “*O sítio da mente*” [1], Dr. Henrique Del Nero procura demonstrar que o problema para se entender como “cérebros” processam passa por desvendar “o código”, ou seja, a forma como neurônios representam internamente “coisas” do mundo (objetos, ideias, etc). Mais que isso, temos que entender como neurônios relacionam tais representações e como isso se traduz em operações computacionais. Ele faz uso de uma alegoria para explicar que neurônios (pulsantes que são) podem codificar informações em “códigos de barra”, dada a semelhança de uma *torrente de pulsos*¹ com o código de barras usado comercialmente. Del Nero perseguiu isso durante sua tese de doutorado [2], e tanto no livro quanto na tese sua inclinação era no sentido de conjecturar que neurônios codificavam informações nas diferenças (ou ângulos) de fases entre pulsos.

Do início da pesquisa ao final desta tese, alguns vetores não mudaram de direção; o mais influente deles é sobre a abordagem *biomimética* e *bioinspirada*², que foi escolhida desde o princípio e mantida ao longo de toda a pesquisa. Nesse sentido,

¹ Em inglês poderíamos dizer um “*streaming*”, um termo que designa um fluxo constante dos potenciais de ação (PA ou *spikes*) gerados pelos neurônios.

² *Biomimética* se vale de imitar soluções que os organismos encontraram para diversos problemas, enquanto uma abordagem *bioinspirada* não tenta imitar, mas sim se inspirar em soluções biológicas. Exemplos de laboratórios de biomimética, em Stanford: <http://www-cdr.stanford.edu/biomimetics/> cujo lema é “*Biomimesis: to mimic life, to imitate biological systems*”. Outro exemplo é o laboratório no MIT: <http://web.mit.edu/sangbae/www/> (acessados em 07/07/2011).

esta tese tem grande incursão em textos clássicos da neurociência [3], [4], [5], [6], [7], entre outros.

Esse esforço em aproximar engenharia da biologia tem promovido avanços em diversos campos de ambas as ciências. Há uma justificativa pela escolha: desde 1943, em quase sessenta anos de simulações de circuitos neurais, vimos vários tipos de redes neurais que, grosso modo, se resumem em sistemas *classificadores*, *preditivos*, *concentradores*, ou em *aproximadores* de funções matemáticas [8], [9]. Até onde esta pesquisa pode avançar, não foi encontrado na literatura descrições de máquinas neurais artificiais que executem algo que se pareça com *processos mentais* ou *pensamentos*; no sentido de cadeia de representações se relacionando causal e hierarquicamente, gerando outras cadeias, novas representações, vetando o andamento de outros processos, etc.; fluxos de operações naturalmente reconhecidos como cognitivos. Mas, todos esses processos acontecem em sistemas nervosos. Assim, no entendimento do autor dessa tese, se quisermos avançar no sentido de gerar máquinas com capacidades “cognitivas”, qualquer que seja o sentido para o termo cognitivo, é necessário revisitar os exemplos biológicos. É preciso tentar, onde for possível, imitar as soluções biológicas, e onde não for possível, nos inspirarmos nelas.

Destaca-se aqui o Dr. Del Nero em virtude de ser ele um personagem no processo pelo qual esta investigação tomou o rumo das redes neurais pulsadas (RNP) e da computação por assembleias neurais. Como foi dito, o objetivo inicial era descobrir como construir robôs e agentes inteligentes, mais cognitivos; portanto, o caminho apropriado foi procurar a disciplina sobre fundamentos de ciência cognitiva, ministrada por ele na EPUSP.

No livro acima citado [1], Del Nero declara algumas vezes que assembleias neurais são importantes para os processos mentais, mas não chega a tentar explicar *como* elas possivelmente operam para representar ou computar nos SNs. Em outras palavras, embora o autor soubesse que assembleias são importantes para várias operações mentais, no momento em que ele desenvolveu sua tese e o livro, começava a surgir tecnologias que permitem investigações mais profundas sobre identificação e computação por assembleias.

Em certo momento havia de minha parte dúvidas sobre certos artigos então recentemente publicados, bem como sobre um conceito controverso que recebia várias críticas, a saber, a noção de *grandmother cell* [10]. O assunto tratado nos artigos e o conceito da “célula da vovó” despertaram meu interesse pela codificação neural, assunto também de interesse do professor Del Nero, o que gerou proveitosas discussões.

Os artigos acima citados, escritos por Quiroga et al. [11], [12], relatam que certos neurônios tornam-se responsivos especificamente quando certo objeto é visualmente apresentado para um paciente. Seria a prova da existência das *grandmother cell*? Na verdade, na mesma edição da *Nature* [13], e posteriormente os próprios autores [14], apresentaram uma interpretação mais plausível, associando o fenômeno ao conceito de *representação esparsa*. Esse assunto mostrava-se instigante e ainda pouco explorado academicamente, especialmente no campo das simulações em redes neurais artificiais pulsadas.

Instigado por esse desafio, esta pesquisa tomou um curso paralelo e diferente da representação por meio da codificação pela fase dos pulsos neurais, originalmente perseguido por Del Nero. Mais ainda, fora do que normalmente se espera, a pesquisa desviou-se ligeiramente também das linhas de pesquisas estabelecidas na Escola Politécnica da USP, quais sejam, a da computação por meio de redes PLL acopladas (*phase-locked loop coupled networks*), uma linha de pesquisa com sólida base de conhecimento na EPUSP, liderada pelo Prof. Dr. José R. C. Piqueira (ver [15], [16], [17]); ou ainda das pesquisas em redes neurais desenvolvidas pelo grupo ÍCONE, no LSI/EPUSP, liderado pelo Prof. Dr. Emílio Del Moral Hernandez (ver [18], [19]). Não é fácil tomar um caminho paralelo e escolher um objeto de investigação novo e sem fundamentação teórica rodeado por conhecimento estabelecido e competente. Aqui entra o orientador dessa tese, Prof. Dr. Marcio Lobo Netto, que dentre suas linhas de pesquisa, como Vida Artificial, Jogos Eletrônicos, Inteligência Computacional, e Ciência Cognitiva, etc. (ver [20]); há uma na qual a mudança de rumo desta pesquisa se encaixou: a neurocomputação.

Esta pesquisa perseguiu o caminho de identificar e modelar o substrato físico onde *um possível tipo de cognição* poderia acontecer, com um objetivo em mente, que seria manter a plausibilidade biológica. A tarefa passa então por encontrar modelos

matemáticos que simulem neurônios e que reproduzam várias características e propriedades dos neurônios naturais, com relativamente baixo custo computacional. Mas, isso em si não foi suficiente. A que tipo de influências as células nervosas e as redes estão sujeitas? Como os grupos de neurônios se organizam? Respostas para estas perguntas demandaram estudos dos efeitos das plasticidades nas células e nas topologias das redes, e a influência da temporização e dos atrasos na propagação dos pulsos neurais nas redes.

Um dos quesitos das redes neurais é reproduzir fenômenos de representação e organização computacional que ocorrem na ordem de centésimos de segundos nas redes biológicas. Tais eventos são muito rápidos para justificarem transformações plásticas nas sinapses dos neurônios; assim, é possível que existam mecanismos nas redes neurais que representem estados de mundo e que organizem execução de operações algorítmicas sem a necessidade de utilizar mecanismos plásticos das células neurais, ou sináptico, mais permanentes (hebbianos).

Surgiu então a ideia das assembleias neurais biestáveis, que reverberam e se auto-organizam que, grosso modo, operam de forma análoga aos flip-flops – elementos biestáveis que agem como unidades de memória em computadores digitais. Enquanto simulações de assembleias biestáveis eram elaboradas, percebeu-se que coalizões podem, ao mesmo tempo, representar o mundo e controlar algoritmos computacionais. Simular esse tipo de comportamento em RNPs tornou-se mais importante do que a própria perseguição sobre os atributos mínimos para a cognição. Formações de assembleias biestáveis podem explicar *um tipo* de estrutura computacional que levou à cognição – mas não se advoga aqui que seja *a forma* como os organismos se tornaram cognitivos.

É provável que a continuação desse trabalho de pesquisa se volte para uma nova forma de computar com neurônios pulsados. Neste tipo de computação, grupos de neurônios pulsados formam assembleias que se ramificam, em paralelo, explorando desfechos para os eventos que as geraram. Como o leitor terá oportunidade de ver, este tipo de formação é capaz de representar estados de mundo (exterior ou interior), de codificar e memorizar signos e eventos, além de permitir a criação de cadeias de assembleias, criando internamente relações causais entre elas. As coalizões podem se ramificar e interagir umas com as outras, podem vetar ou

desarmar outros ramos; enfim, podem realizar aquilo que chamamos de algoritmo computacional paralelo.

O leitor está convidado a seguir um caminho um tanto longo, cheio de ideias e argumentos de várias disciplinas, sem tantas fórmulas matemáticas como é comum numa tese da engenharia; mas que ao final, espera-se, terá justificado o surgimento de uma nova forma de computação com redes neurais pulsantes.

A curiosidade científica pelas coalizões neurais impeliu essa pesquisa no sentido de identificar como, possivelmente, as assembleias computam; como elas podem controlar fluxos ininterruptos de impulsos nervosos, e como representam objetos, ideias, ações; na esperança de assim identificar “o código” de que falava Del Nero. O professor Del Nero, que não perdia oportunidades para nos “acicatar”, verbo que ele gostava, não está mais conosco. Ficam as lembranças, as discussões, as mudanças de rumos que ele promoveu, e ao final, uma proposta sobre “como” as assembleias de neurônios podem orquestrar operações computacionais. Computação por coalizões neurais, a computação pelas coincidências, esse é o assunto que reservam ao leitor as páginas seguintes dessa tese.

1.1 Objetivos

O objetivo principal desta tese é investigar, e fundamentar teoricamente, *como* assembleias de neurônios pulsantes realizam operações computacionais. Para isso, temos que lançar metas parciais que, ao final, nos permitam compor um panorama que explique, numa abordagem funcional, como ocorrem as coalizões, o que elas significam para a máquina computacional (a rede), e como isso resulta em respostas comportamentais.

Algumas metas parciais devem ser:

- (1) quais tipos de neurônios são apropriados para simulação de assembleias;
- (2) quais mecanismos de plasticidade celular, topológica e sináptica são influentes para assembleias;
- (3) como se formam as assembleias;

- (4) como as assembleias representam estados de mundo (internos ou externos);
- (5) como as coalizões controlam fluxo contínuo de informações;
- (6) como isso resulta em execução de processamento de informação (*algoritmo*) e resposta comportamental.

Abordagem funcional quer dizer que o objetivo principal será descrever “um possível” modo de operação pelo qual assembleias realizam *computação* (serial e paralela); numa investigação de possibilidades teóricas, preocupada com a coerência das descrições sobre um modo de operação das coalizões.

1.2 Computação não-padronizada e algoritmo

Cientistas da computação possivelmente discordarão do uso informal de alguns termos nesta tese, em particular dos conceitos de *computação* e *algoritmo*. O termo *computação* é usado neste texto num sentido amplo, não-padronizado, como no livro “*Non-standard Computation: molecular computation - cellular automata - evolutionary algorithms - quantum computers*” [21]. Computadores (no sentido formal da máquina de Turing) são máquinas que manipulam dados sequencialmente de acordo com um conjunto de instruções. No senso estrito das máquinas sequenciais (e.g. Turing machines, Church λ -calculus, etc), computação é uma atividade relacionada ao uso de computadores para resolver problemas executando algoritmos. Em matemática e ciência da computação, computar é o ato de executar uma atividade orientada por meta que requer ou tira proveito de um computador. Ocorre que os termos computador e computação foram antes usados como sinônimo para “tratamento de dados” orientados para um fim.

Por exemplo, no livro “*Biophysics of Computation: Information Processing in Single Neurons*” [7], o autor inicia o livro com a frase: “*O cérebro computa! Este é um truísmo aceito pela maioria dos neurocientistas [...]*”. Fato é que o termo *computação* tem sido usado para sistemas que recebem dados e aplicam sobre eles operações com o propósito de obter um resultado. É nesse sentido que se toma a liberdade de utilizar o termo *computação* neste trabalho. Por conseguinte, o termo *algoritmo* também é usado aqui num sentido mais amplo, não no sentido dos algoritmos das

máquinas de Turing. *Algoritmo* aqui deve ser entendido como um conjunto de operações (geralmente sobre dados, eventos, impulsos nervosos, representações) com o propósito de realizar uma tarefa, um processo, um tratamento qualquer, e assim gerar um resultado.

1.3 Motivação e justificativa

A motivação principal da pesquisa é identificar possíveis formas de criar redes neurais pulsadas que *computem* utilizando o fenômeno de formação de coalizões como base. Até o momento, não são frequentes as pesquisas em redes neurais que abordem especificamente este assunto. Como mostram Paugam-Moisy e Bohte [22]: “The current consensus agrees that cognitive processes are most likely based on the activation of transient assemblies of neurons [...], although the underlying mechanisms are not yet understood well.”

A grande motivação é tentar explicar, ao menos teoricamente, uma das possibilidades pela qual as assembleias realizam operações como representar coisas, disparar ou vetar certas respostas estereotípicas, memorizar estados de mundo e estados internos, e executar algoritmos que resultem em comportamento inteligente, quem sabe cognitivo.

Justifica-se a pesquisa neste sentido, acima de qualquer coisa, porque se trata de investigar e tentar entender como funcionam os sistemas nervosos biológicos. Se alguma parte do que se relata nesta tese for útil para conseguir simular em computadores partes dos circuitos neurais existentes em organismos, podemos dar um passo no sentido de criar máquinas que ajudem a entender o processamento nos encéfalos.

Contudo, se o que esta pesquisa revela não se observar em organismos, ao menos uma nova forma de projetar dispositivos puramente computacionais em redes de neurônios pulsantes será proposta.

Em ambos os casos, abre-se uma avenida para novas pesquisas em redes neurais pulsadas com uma nova visão: a introdução dos conceitos de eletrônica digital e de projetos de circuitos sequenciais na concepção e design de redes neurais.

1.4 Revisão da literatura

O objetivo do tópico de revisão bibliográfica é mostrar a base de conhecimento consultada para o projeto, ou seja, um levantamento do estado da arte das publicações na área da pesquisa; que ao final, é uma experiência subjetiva. Esta tese é resultado de um trabalho multidisciplinar. No decorrer do texto o leitor encontrará indicações sobre como obter informações sobre os temas pontualmente. Assim, nesta sessão, serão descritas apenas as grandes classes de literatura consultada, como um mapa para os primeiros passos em cada assunto.

A primeira classe de leituras tem a ver com neurociências. Os livros e textos básicos nessa área, indicação nossa: “*Principles of neural science - Fourth Edition*”, Kandel, Schwartz e Jessel [3]; ou “*Neuroscience - Third Edition*”, Purves et al. [5]; ou em português, o livro “*Neurociências: Desvendando o Sistema Nervoso - 3a. Ed.*”, Bear, Connors e Paradiso [6]. Há ainda uma enciclopédia excelente, com verbetes curtos e objetivos, geralmente escritos pelas maiores autoridades em cada assunto: “*The Handbook of Brain Theory and Neural Networks 2nd Ed.*” Arbib, M. A. [4]. Com esses livros é possível ter bom conhecimento sobre neurônios biológicos, suas funções, geração de potenciais de ação, propagação desses potenciais, sinapses, etc.

A segunda classe de textos está relacionada a simulações de redes neurais. Indicamos como leitura introdutória o texto “*Computing with spiking neuron networks*”, de Paugam-Moisy e Bohte [22], disponível na internet, que é um capítulo da primeira edição do livro *Handbook of Natural Computing*. Há um sem número de revisões sobre o tema, um bom exemplo é “*Simulation of networks of spiking neurons: A review of tools and strategies*”, Brette et al. [23], ou o importante artigo de Maass “*Networks of spiking neurons: The third generation of neural network models*” [24]. Livro clássico em redes neurais, “*Neural networks: a comprehensive foundation; 2nd ed.*”, Haykin-98 [8], é essencial para aqueles que querem trabalhar com redes neurais tradicionais.

Outra classe de textos se refere à modelagem neural. Um livro que está se tornando clássico sobre o assunto é “*Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*”, Izhikevich [25]. O livro mostra como neurônios podem ser tratados como sistemas dinâmicos, e descreve o processo matemático pelo qual se

chega ao modelo simples de Izhikevich. Também desse autor, um texto essencial é *“Which model to use for cortical spiking neurons?”* [26], que mostra os tipos de atributos necessários para simulação de neurônios pulsantes (*spiking neurons*) além de comparar vários modelos de neurônios artificiais; além de *“Simple model of spiking neurons”* [27]. Outro livro essencial sobre neurônios pulsantes é *“Spiking Neuron Models: Single Neurons, Populations, Plasticity”* de Gerstner e Kistler [28].

Uma quarta classe está relacionada com plasticidade. Os livros de neurociência anteriormente citados não discutem as últimas descobertas de pesquisas, eles estão mais focados em conhecimento estabelecido. Assim, a fonte para atualização em assuntos como plasticidade, ou assuntos correlatos, são os artigos em periódicos. Em plasticidade temos uma subclasse que trata de plasticidade estrutural, e os textos nos quais o leitor poderá se familiarizar mais rapidamente com o assunto são: *“Anatomical and Physiological Plasticity of Dendritic Spines”* de Alvarez e Sabatini [29]; *“Cortical rewiring and information storage”*, de Chklovskii, Mel e Svoboda [30], e *“Cell Type-Specific Structural Plasticity of Axonal Branches and Boutons in the Adult Neocortex”*, Vincenzo De Paola et al. [31].

Outra subclasse do assunto plasticidade é a plasticidade homeostática, nesse sentido o leitor se beneficiará com os textos *“Homeostatic plasticity in the developing nervous system”* de Turrigiano e Nelson [32], *“Unraveling Mechanisms of Homeostatic Synaptic Plasticity”* de Pozo e Goda [33], e ainda *“Strength through Diversity”* de Nelson e Turrigiano [34].

Há ainda a subclasse de plasticidade de curto termo, ver *“Synaptic modification by correlated activity: Hebb's postulate revisited”*, de Bi e Poo [35], e *“Synaptic plasticity: taming the beast”*, de Abbott e Nelson [36]. E por último, temos a subclasse de leituras sobre “spike-timing dependent plasticity” (STDP), que o leitor se familiarizará lendo *“Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs”*, de Markram et al. [37]; sobre uma perspectiva histórica ver *“Dendritic Excitability and Synaptic Plasticity”* de Sjöström et al. [38]; uma descrição cronológica de fatos marcantes sobre plasticidade em *“Phenomenological models of synaptic plasticity based on spike timing”*, de Morrison, Diesmann e Gerstner [39]; ou ainda *“Plasticity in single neuron and circuit computations”*, de Destexhe e Marder [40].

Sobre computação dendrítica e ajuste temporal entre neurônios, a bibliografia consultada foi: “*Biophysics of Computation*”, Koch, [7]; “*Dendritic Computation*”, London & Häusser, [41]; “*Dendritic Processing*” Segev & London, no “*The Handbook of Brain Theory and Neural Networks 2nd Ed.*” [4]; “*Pyramidal neurons: dendritic structure and synaptic integration*”, Spruston [42]; “*Dendritic Excitability and Synaptic Plasticity*”, Sjöström et al. [38]. Sobre propagação e atrasos de pulsos nas árvores dendríticas, um dos melhores textos é “*Propagation of action potentials in dendrites depends on dendritic morphology*”, de Vetter, Roth e Häusser [43].

Há ainda uma série de conceitos ligados à formação de assembleias neurais, como *synfire chain*, que o leitor encontrará explicações em “*Synfire chains*”, de Abeles [44], ou a noção de *synfire braid* no texto “*A model of neocortex*” de Bienenstock [45], e ainda a noção de grupos *policronizados*, no texto “*Polychronization: Computation with Spikes*”, de Izhikevich [46] ou ainda “*Polychronous Wavefront Computations*” de Izhikevich e Hoppensteadt [47].

Como foi dito, essa é uma indicação subjetiva, pode acontecer de o leitor preferir outros textos que expliquem melhor estes conceitos e fenômenos acima explicados, neste caso, sugere-se que o leitor faça uma busca aprofundada em outros textos e referências que estão citadas nos textos acima. Mais indicações sobre leituras de temas pontuais são apresentados durante o discurso da tese.

1.5 Publicações associadas a esta tese

A esta tese estão associadas as seguintes publicações:

- Um dos artigos diretamente ligado a esta tese é intitulado “*Bistable Memory and Binary Counters in Spiking Neural Network*”, apresentado no congresso: IEEE Symposium on Foundations of Computational Intelligence. FOCI 2011, Paris, França [48]. Nele são descritas as simulações das assembleias funcionando como memórias biestáveis, contadores digitais e como contadores de pilha. Esse artigo serve de base para o capítulo VI desta tese.
- Outro artigo, submetido ao periódico IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, com o título “*Computing with Spiking Neural*

Assemblies”, está em revisão na data do fechamento da tese [49]. Nele apresentamos uma visão geral da abordagem de computação por assembleias, analisamos um algoritmo demonstrando como um agente pode computar respostas estereotipadas complexas por meio de assembleias neurais. Parte desse artigo está publicada aqui, disseminada nos capítulos dessa tese.

- Ainda no começo da investigação, quando nossa pesquisa estava com grande enfoque em cognição, submetemos um artigo ao congresso *Brain Inspired Cognitive Systems-2008*, que resultou num capítulo do livro “Brain Inspired Cognitive Systems 2008”. Artigo e o capítulo têm o título: “*On Building a Memory Evolutive System for Application to Learning and Cognition Modeling*” [50]. Nos textos analisamos a possibilidade de uma rede neural evoluir a partir de construções que se baseiam em Teoria das Categorias, sem envolvimento de coalizões.

1.6 Organização da tese

Após o primeiro capítulo introdutório, é feita, no capítulo **II** desta tese, uma revisão sobre os problemas de computação neural, restringindo ao nível de investigação das células neurais. É apresentado um breve histórico das tentativas de simulação neural, das gerações de redes neurais, com destaque para as redes neurais pulsadas; e dentro desse enfoque, como neurônios pulsantes podem “codificar” informações. Neurônios naturais possuem vários comportamentos dinâmicos, assim, discute-se a importância dos modelos matemáticos que representam seus comportamentos, levando em conta o custo computacional de cada modelo e sua proximidade com a realidade biológica.

Neurônios são células bastante flexíveis, apresentam uma série de plasticidades que podem ser provocadas por processos fisiológicos, por atividade e excesso ou a falta dela, pela presença ou ausência de moduladores (neurotransmissores), entre outros. Plasticidades são discutidas no capítulo **III**, assim como as influências desses mecanismos na computação geral da rede, e prováveis influências nas coalizões. Discute-se ainda como modelar todos esses mecanismos plásticos em simulações de assembleias, analisa-se a questão da computação dendrítica, e ajuste do tempo de retardo na propagação de sinais entre assembleias.

Isolados, neurônios computam pouco; é em rede, operando em grupos, que as células apresentam uma extraordinária capacidade computacional. Redes é o assunto do capítulo *IV*, onde se discute alguns tipos de configurações e estratégias para simulações de redes neurais, a computação em pequenos grupos de neurônios, os circuitos marca-passo e centrais geradoras de padrão, avançando então para como as assembleias temporais se formam. Ao final do capítulo, discutimos o que são as coalizões biestáveis e como elas podem funcionar como memórias de estados de mundo, estados internos, entre outras coisas.

No capítulo *V* é apresentada uma proposta sobre como as assembleias executam algoritmos, que podem ser deveras complexos. Os conceitos de ramificação e dissolução de coalizões, representação e persistência, funções lógicas executadas na formação de assembleias (funções pseudo-booleanas), as relações causais e hierárquicas entre as coalizões são apresentados nesse capítulo. Porém, a proposta da computação por assembleias possui problemas que devem ser observados neste momento nascente da pesquisa; assim, ao final do capítulo discutem-se os principais problemas a serem enfrentados pela abordagem da computação por coalizões, e algumas possíveis soluções.

No capítulo *VI* são mostrados os primeiros resultados de simulações executadas para validar a proposta teórica aqui apresentada. Foram realizadas simulações para testar basicamente o cerne da proposta de computação por assembleias que reside na execução de dois grupos de funções: as pseudo-booleanas, e as assembleias biestáveis. Assim, o capítulo mostra os resultados obtidos.

O capítulo *VII* apresenta as conclusões e discute as possibilidades para novas pesquisas e trabalhos futuros usando essa abordagem.

1.7 Hipóteses, premissas e conjecturas que guiam a tese

A seguir será apresentado um resumo comentado das hipóteses, premissas e conjecturas mais importantes, que embasam esta tese. Propositamente, apenas as hipóteses, premissas e conjecturas que servem como linhas-guia da abordagem da computação por assembleias serão citadas.

1: Neurônios utilizam temporização entre disparos como elemento para realizar uma série de operações computacionais.

Falamos em *neurônios* cientes dos diversos tipos de comportamento que eles apresentam. Ainda assim, há razões para acreditar que a maioria das respostas neurais está relacionada com alguma forma de computação temporal, cujo elemento principal é o intervalo entre disparos dos neurônios.

2: Os atrasos na propagação dos disparos entre neurônios também são componentes importantes no cenário da computação em redes neurais pulsadas.

Se a temporização entre disparos é um fator importante, então os atrasos axonais, sinápticos e na árvore dendrítica devem ser considerados. As relações temporais entre disparos neurais causam a formação de coincidências nos disparos.

3: Os tempos entre disparos e os atrasos fatalmente levam os neurônios a dispararem coincidentemente, formando grupos que disparam temporalmente “travados”.

Ao fenômeno de disparo conjunto, em que neurônios formam grupos que pulsam em janelas de tempo efêmeras, damos o nome de *assembleias*. Durante a tese usamos ainda os termos *coalizões*, ou *eventos*, ou ainda *grupos policronizados*, para descrever estes fenômenos.

4: Além da temporalidade, assembleias de neurônios se formam porque existem fatores de acoplamentos (pesos sinápticos) diferentes interconectando-os.

Em outras palavras, não são apenas as relações temporais que provocam coalizões. É preciso que os neurônios tenham conexões físicas e pesos sinápticos adequados para que a propagação entre eles resultem nas assembleias.

5: Para formar ou dissolver uma assembleia, inevitavelmente, algum tipo de função lógica (E, OU, NÃO) deverá ser executada.

A formação/dissolução de assembleias é resultado de algum desses eventos: a conjunção de disparos (disparos de um grupo *E* de outro devem coincidir), a disjunção de disparos (disparos de um *OU* outro grupo geram novos eventos), e a negação do evento (a inibição de uma assembleia equivale à função lógica NÃO). Associação entre esses resultam nas outras funções lógicas conhecidas.

6: Assembleias podem disparar outras assembleias, formando ramificações.

Esse fenômeno é a gênese de qualquer *algoritmo* realizado pelas coalizões. Nesse caso, cada novo ramo deve ser considerado um novo evento em paralelo, que pode também ser pensado como uma *opção* no tratamento de informação. Inibição ou veto das ramificações bloqueiam certas opções. O resultado ao final da cadeia ramificada é a operação computacional, o algoritmo, realizado pelas coalizões.

7: Assembleias podem retro-alimentar outras assembleias formando laços biestáveis.

Essa operação pode reter um bit de informação, mas geralmente ela memoriza mais que isso, porque é uma forma de representação interna do evento que causou a construção daquela assembleia biestável.

8: Neurônios podem compartilhar seus tempos de disparo e participar, isoladamente, de mais de uma assembleia.

Isso eleva enormemente a capacidade computacional das redes. A quantidade de assembleias nas quais um neurônio pode participar será sempre finita (ver [51]); porém, a quantidade de combinações para formação de coalizões esparsamente distribuídas será potencialmente infinita.

9: Uma vez que assembleias formam laços biestáveis e executam funções lógicas básicas, assim, teoricamente, elas podem realizar qualquer tipo de “algoritmo” por meio de ramificações e inibições de outras assembleias.

Dessa forma, coalizões seriam basais na criação de um tipo de computação nos SNs: a computação sequencial, na qual relações causais e hierárquicas são imprescindíveis e estão implícitas nas formações das assembleias.

As hipóteses acima deverão ser comprovadas na tese, uma vez que constituem o cerne da abordagem de computação baseada em coalizões. Existem outras hipóteses que deverão ser investigadas como parte do corpo teórico, sem assumir o compromisso de comprová-las nesta tese.

10: As coalizões requerem estruturas organizadas para acontecerem. Computação por assembleias não é um fenômeno puramente emergente; ao contrário, deve ser resultado de uma construção estabelecida de relações espaço-temporais entre neurônios.

Isso quer dizer que nessa abordagem entendemos que não basta colocar um grupo de neurônios conectados aleatoriamente e esperar que as assembleias realizem computação. Ao contrário, é resultado de uma construção elaborada.

11: Neurônios se ajustam, ajustam o fator de conexão entre si, e ajustam suas próprias dinâmicas e a dinâmica entre a vizinhança, a fim de manter o cérebro operacional.

Sabe-se da neurociência que neurônios são plásticos de várias formas. A hipótese aqui levantada é que tais plasticidades influenciam na formação de assembleias, na manutenção delas, e no aprendizado que nelas acontecem. Apesar de prever e incluir plasticidade no modelo de computação aqui desenhado, não será objetivo nessa tese a comprovação dessa hipótese, ficando para futuras investigações.

12: Assembleias se formam como resultado da relação temporal associada a fatores de acoplamento entre grupos. O componente de acoplamento, a plasticidade sináptica, é assunto conhecido e bastante estudado em neurociência. Da mesma forma, deve haver um correlato de plasticidade temporal que permite algum tipo de aprendizado temporal nas redes que computam por meio de formação de assembleias.

Essa é uma hipótese que resulta de inferência, porém que deve ficar registrada para investigação futura. Trata-se de uma proposição elementar, mas que tem ganhado pouca atenção na investigação científica. Se a abordagem das coalizões estiver correta, como é argumentado e mostrado nesta tese; então isso abre perspectivas para investigação sobre um novo tipo de “aprendizado”, que acontece por meio de ajuste nas relações temporais entre neurônios isolados ou entre grupos.

Dessa forma, além do aprendizado “hebbiano”, a abordagem da computação por coalizões traz implícita a possibilidade de ocorrer também um tipo de aprendizado, de ajuste na rede, baseado no tempo e no ajuste das relações temporais.

Em suma, neurônios das assembleias se adequam temporalmente a fim de cooptarem novos neuroides e fortalecerem as assembleias das quais participam, num processo de auto-organização.

2 COMPUTAÇÃO NEURAL

Parafraseando Koch [7], sistemas nervosos computam. Isto é um truísmo que os neurocientistas têm como certo. Sistemas Nervosos (SN)³, designando o “sistema computacional biológico” como um todo, são complexos formados por vários tipos de tecidos e células, divididos em várias subestruturas. Há certo consenso sobre os *neurônios* serem as unidades responsáveis pelas operações de processamento de informações que ocorrem nos encéfalos. Questiona-se o papel das células *glias* na computação cerebral; porém, apesar de haver mais células *glias* que neurônios nos SNs e de elas favorecerem certos tipos de operações, até onde se sabe, elas não estão diretamente ligadas ao processamento de informação, e sim à tarefa de suporte aos neurônios e formações de estruturas cerebrais [3], [6], [5]. Por outro lado, as células neurais isoladas não são capazes de explicar o complexo comportamento dos sistemas nervosos.

2.1 Níveis de investigação cerebrais

Há vários níveis de investigação aplicáveis a pesquisas do funcionamento dos SN e encéfalos; porém, o conceito de *neurônio* é central. O que o diferencia das demais células é sua capacidade de gerar e transmitir sinais elétricos a distâncias relativamente longas. A partir do conceito de neurônio é possível descer ao nível da biofísica neural, ou mais, até a biologia molecular e a regulação do funcionamento celular por meio dos genes. Esses dois níveis de investigação explicam as transformações que ocorrem na célula neurônio, que modificam seus padrões de resposta e seus comportamentos quando expostos a certos regimes de operação. Em outras palavras, ocorre um tipo de modificação intrínseca aos neurônios, de uma

³ Usa-se aqui *Sistemas Nervosos (SN)* no sentido de designar a “máquina neural” como um todo. O nome do centro do sistema nervoso humano é *encéfalo*, composto por três partes principais: o grande cérebro, o tronco encefálico, e o cerebelo [6]. Contudo, a noção de que toda a computação ocorre no encéfalo tem perdido força, porque várias operações ocorrem localmente, distribuídas pela medula espinhal, gânglios e redes localizadas. Acredita-se que processamentos sejam distribuídos pelo corpo dos animais (vertebrados e invertebrados), abordagem que ganha o nome de *cognição corporificada* ou incorporada (*embodied* ou *situated cognition*). Ver “*Learning and Memory Without a Brain*” em [136]; ou “*Embodied Cognition: A field guide*”, [137].

forma geral referida como *plasticidade*, que é responsável por modificar o comportamento da célula. Investigações nesse campo são chamadas de *plasticidade homeostática* ou *intrínseca* (ver [32], [34], [33]). Neste nível de investigação o cientista se preocupa com o fato de que o elemento principal na computação cerebral se modifica dependendo das condições nas quais opera.

Acima do nível das células podemos destacar o nível dos circuitos neurais e das interconexões entre neurônios (redes e topologias). Neste nível, novos fatores passam a influir na máquina computacional, dos quais devemos destacar quatro fatores: o *grau de acoplamento*, a *modulação* das interconexões, os *atrasos* de propagação dos sinais na rede e a *localização* das conexões na árvore dendrítica.

O primeiro fator é o grau de acoplamento entre neurônios. Neurônios podem se interconectar de duas maneiras: as sinapses elétricas e as químicas. As primeiras (*gap junctions*) são conexões elétricas que permitem trocas de correntes iônicas de forma direta e praticamente instantânea (ver [52], [53], [3]). Para um valor de zero a um, indicando o grau de acoplamento das conexões, as conexões elétricas teriam o valor máximo; ou seja, conexões elétricas têm “peso sináptico” unitário, porque o acoplamento é direto e total. Já o segundo tipo de conexão é mediado por componentes químicos (monoaminas, peptídeos, etc.), genericamente denominados neurotransmissores. Estes, e os respectivos canais receptores, *amplificam* ou *atenuam* sinais propagados entre neurônios (ver [3], [6]). O fato de as sinapses químicas modificarem o grau de acoplamento entre neurônios é um fator relevante para a máquina computacional como um todo. As regras pelas quais as modificações no acoplamento acontecem começam a ser desvendadas, e ganham o nome genérico de *plasticidade sináptica* (ver [39], [40]).

Mais que modificar o grau de acoplamento ou o peso da sinapse, neste nível de investigação deve-se analisar um fato importante: neurotransmissores têm um efeito “modulador”, ou seja, intensificador ou redutor das respostas neuronais. A presença ou ausência de certos neurotransmissores em sistemas biológicos pode alterar drasticamente o comportamento global da rede; digamos, modificando fatores operacionais e respostas individuais dos neurônios. Segundo Bear e colaboradores, “[...] mudanças químicas podem deixar traços duradouros em sua história, drogas podem alterar o balanço do potencial sinalizador, e, em um sentido literal, o encéfalo

e suas substâncias químicas são uma coisa só” [6] pp.165. Essa frase mostra a importância dos moduladores na computação global. Contudo, apesar da importância dos neurotransmissores para os SNs, especialmente para encéfalos, uma modelagem do seu efeito sobre a computação em redes neurais artificiais (RNA) é complexa, demanda conhecimento ainda não totalmente dominado.

Um exemplo de simulação de moduladores é feita em [54], em que se aborda o problema da recompensa distante. Em condicionamentos pavloviano e instrumental, recompensas chegam segundos, minutos ou horas após as ações compensatórias terem sido executadas; criando um enigma conhecido como "*distal reward problem*": como o cérebro sabe quais padrões de disparo, advindos de quais neurônios, foram responsáveis pela recompensa se (1) o padrão não está mais lá quando chega a recompensa, e se (2) todos os neurônios e sinapses forem ativados durante o período de espera da recompensa? Izhikevich apresenta em [54] uma solução para explicar o problema introduzindo um modulador à base de dopamina que simula cinéticas lentas de plasticidade sináptica sensíveis à concentração de dopamina extracelular durante o período de poucos segundos. Interessante, mas o mecanismo simula apenas o efeito de dopamina isolada (há mais de 200 neurotransmissores) atuando especificamente como um modulador de plasticidade. A introdução de simulações com neurotransmissores ficará como proposta para trabalhos futuros, sob demanda, e quando a base de conhecimento sobre o assunto for mais sólida.

Ainda no nível das interconexões neurais, outro fator a ser considerado é o atraso que sinais elétricos sofrem ao percorrer o axônio do neurônio emissor, a sinapse química, e a árvore dendrítica do neurônio receptor até chegar ao seu soma. Como será mostrado, este fator é determinante em computação por assembleias. Embora seja negligenciado em várias simulações de RNAs, na computação por assembleias o retardo na propagação de sinais é fator protagonista, tendo a mesma importância que o peso sináptico das interconexões neurais.

Há ainda o fator *posição* das conexões na árvore dendrítica que deve ser considerado no nível das conexões. Em poucas palavras, a posição das sinapses na árvore dendrítica tem sido apontada como responsável por inúmeras funções computacionais (ver tópicos 3.6 e 3.7 nesta tese, ou *Dendritic Processing* em [4], ou ainda [7] cap. 5).

Estudos recentes mostraram que a morfologia, a posição das conexões sinápticas, a multiplicidade de conexões, e a fisiologia das árvores dendríticas afetam de forma relevante o processamento que o neurônio executa. Por exemplo, dendritos não são estritamente passivos, e na interação das sinapses com os dendritos estes podem apresentar propriedades como atenuação ou controle de ganho, filtro, inibição ou “veto local” de ramos da árvore, integração espaço-temporal, detecção de coincidência, entre outras (ver figura 3.4). Na medida do possível este fator será considerado na abordagem de computação por assembleias nesta tese.

Saindo do nível das conexões neurais, “subindo” um nível, encontramos a camada dos subsistemas formados pelas redes neuronais. Como exemplo, pode-se destacar alguns núcleos neurais que executam funções de gerações de padrão de disparo (CPG⁴). Além desse, há a camada das estruturas corticais; e ainda o nível dos encéfalos como um todo. Neste nível das estruturas e dos sistemas é que se investiga a “conduta social” dos neurônios, suas coalizões, os sincronismos, as operações computacionais que emergem da interação entre populações e que neurônios isolados não conseguem apresentar. Acima do nível de investigação do encéfalo e suas estruturas é possível estudar o comportamento elaborado, inteligente, a cognição e funções mentais, todos esses com alto nível de abstração, níveis onde acontecem as mais sofisticadas operações computacionais.

Nesta tese, no contexto da engenharia, o escopo da análise ficou delimitado, no nível celular, aos tipos de *computação* executados por neurônios isolados e a algumas de suas propriedades de plasticidade intrínseca. No nível topológico, a algumas regras que modificam estruturalmente as redes. E no nível da interação social entre neurônios, às suas associações em pequenas populações, ou seja, como os neurônios interagem para formar assembleias e a computação que daí emerge. Dessa forma, três níveis dos acima citados são tratados nos modelos simulados na tese: (1) o nível dos modelos que representam os neurônios (nível celular), que são as equações matemáticas que descrevem o comportamento

4 Optou-se por utilizar o termo CPG (Central Pattern Generator), mantendo a sigla em inglês por ser amplamente conhecida e difundida tanto nas ciências exatas (computação neural) quanto na neurociência e biologia. Os CPGs serão analisados no tópico 4.2.1 desta tese (ver revisões em [116], e [120]).

neural. (2) O nível das interconexões, que trata a topologia da rede, os atrasos de propagação, os pesos sinápticos entre conexões neurais, e em algum grau, a posição das conexões na árvore dendrítica, ficando a modelagem da modulação feita por neurotransmissores como proposta futura. (3) O nível de subsistemas (das assembleias ou coalizões) que realizam operações computacionais num grau de abstração superior ao que as estruturas físicas dos neurônios interconectados conseguem mostrar.

Como já foi dito, procurou-se nesta tese simular fenômenos biologicamente plausíveis, usando uma abordagem ora *bioinspirada*, ora *biomimética*. Assim, para representar neurônios procurou-se usar modelos matemáticos plausíveis em termos biológicos; ou seja, modelos que representem o fenômeno celular de sorte que reproduzam a computação que os neurônios executam o mais fielmente possível. Por isso, a modelagem deve levar em conta, entre outras coisas, as transformações intrínsecas da célula (plasticidade homeostática). Dessa forma, nos tópicos seguintes serão analisados primeiramente os fatores do comportamento dos neurônios individuais (nível celular), a fim de justificar a escolha dos modelos matemáticos usados nas simulações das redes neurais posteriormente.

2.2 Neurônios e neuroides⁵

Na literatura clássica, as células nervosas são anatomicamente divididas em três partes: *dendrito* ou árvore dendrítica, *soma* ou o corpo celular e *axônio* ou árvore axônica. Estes últimos são, respectivamente, os responsáveis por gerar e propagar potenciais de ação (PAs), ou *spikes*⁶. A descrição de células nervosas, dos canais iônicos e dos mecanismos de equilíbrio no potencial de membrana, da despolarização e da geração de PAs em células nervosas, das conexões sinápticas, das trocas de sinais elétricos e da influência entre neurônios são todos assuntos

⁵ Usou-se nesta tese o termo neuroide para diferenciar a referência à célula biológica do neurônio. Assim, o termo neuroide se refere a uma equação matemática que tenta emular o fenômeno computacional que o neurônio realiza no domínio biológico (ver explicação no texto).

⁶ *Spike* é um termo em inglês para o potencial de ação. Como é um termo curto e descreve um fenômeno específico na natureza, às vezes é mais prático usá-lo.

amplamente conhecidos, ou podem ser facilmente encontrados na literatura de neurociência; assim, características, propriedades e mecanismos das células nervosas não serão descritos em detalhes neste trabalho, devendo o leitor recorrer à literatura apropriada: [3], [6], [5], [7], [55]. É pertinente aqui, questões sobre como (e o que) os neurônios (individualmente) computam.

Podemos dizer que cada uma das partes anatômicas dos neurônios realiza algum tipo de processamento computacional e a integração desses processamentos resulta em um sistema dinâmico não linear. O tratamento dos neurônios como sistemas não lineares tem sido constante na literatura científica recente (ver [25], [23], [56], [57], e referências).

Neurônios isolados são incapazes de explicar a existência de *inteligência* e *cognição*, no senso comum desses termos. Células nervosas interconectadas recebem o nome de *rede*, uma metáfora ao tecido nos quais os nós são diretamente conectados aos nós vizinhos e adjacentes. Ainda que a alegoria seja válida, agrupamentos de neurônios em nada lembram uma rede de pescador ou a malha de um alambrado, por exemplo.

Em mamíferos, células motoras da medula espinhal, com modesto número de dendritos, podem fazer 10.000 conexões (2.000 no corpo celular e 8.000 nos dendritos) [3] p. 25. No córtex de ratos, células nervosas recebem em média 7.800 sinapses ao longo de 4 mm de dendritos [7] p. 87; e em cerebelos as células de Purkinje podem fazer mais de 100.000 conexões [58], [3]. Assim, a forma como os neurônios são interconectados – a *topologia da rede* – é de extrema importância no resultado final da computação que se pretende executar.

Em suma, simular neuroides com a mesma capacidade dos neurônios é uma tarefa bastante complexa por vários motivos: cada unidade computacional é um sistema dinâmico não linear, as unidades se modificam em função da atividade a que estão sujeitas, cada unidade forma um número enorme de conexões e interações com outras unidades, para citar apenas três motivos. Dessa forma, vamos nos atentar primeiro aos detalhes importantes para simulação dos neurônios.

2.3 Modelando neurônios

Investigações e resultados experimentais advindos da neurociência têm alavancado o desenvolvimento das redes neurais pulsadas (RNPs) que são baseadas em modelos neurais que simulam disparos de *potenciais de ações* (PA). A fim de diferenciar do fenômeno biológico PA, neste trabalho optou-se por usar o termo *spike* para designar pulsos gerados por neurônios artificiais em RNPs, e reservar o conceito PA apenas para denotar o fenômeno biológico. Dessa forma, nesta tese, *spike* se refere a pulsos gerados em simulações, e PA se refere ao fenômeno biológico de disparo neural.

RELEVÂNCIA PARA A TESE:

Os comportamentos dos neurônios são ricos e complexos. Mostra-se a seguir que os primeiros modelos matemáticos de neurônios e de RNAs simplificavam, às vezes em demasia, tais comportamentos. Isso não quer dizer que as RNAs tradicionais sejam menos importantes. Elas têm várias aplicações em engenharia e computação. Elas tentam tirar proveito de certas capacidades observadas em cérebros: resolver problemas difíceis de serem resolvidos por meio de algoritmos.

Por outro lado, seguindo a denominação usada por Cruse [59], daqui por diante, usa-se o termo *neuroide* para designar neurônios artificiais pulsados, diferenciando-os dos *neurônios artificiais*, termo que será usado para denominar os modelos neurais das primeiras gerações de redes neurais artificiais (RNA), e.g. *perceptron*; reservando assim o termo *neurônio* apenas para as células biológicas.

Como aponta Arbib, o que é perigoso no sintagma nominal “o *neurônio*” é justamente o artigo “o” [4]. Biólogos estudam inúmeros tipos de neurônios em várias espécies animais e há diferenças enormes nas respostas e comportamentos, na morfologia, e na dinâmica dessas células.

Em teorias e simulações em computadores os neurônios são abstraídos e sofrem simplificações a fim de facilitar o entendimento de diferentes aspectos de seus comportamentos, ou para simplificar as redes neurais em termos de topologias, aprendizado, funções etc. É possível criar uma descrição matemática precisa e unificada das propriedades de uma célula nervosa?

Figura 2.1 – **Comportamento neural**. Vinte respostas experimentalmente observadas em neurônios naturais, cada uma pode representar um comportamento ou tipo de processamento executado. É comum pensar que neurônios somam PEPS e disparam ao atingir um limiar, mas esta é a resposta do *integrador* (L), que é antagônica à do *ressonador* (K). Já a resposta “*rebound spike*” (M) dispara PAs mesmo recebendo pulsos inibitórios. Veja também que (S) e (T) disparam depois que a corrente I de excitação é retirada. Figura adaptada da versão original, de uso livre, cuja autorização de uso pode ser obtida em <http://www.izhikevich.org> - acesso em 20/08/2011.

A maioria dos livros descreve neurônios como *integradores* com *limiar*. Nestas descrições, neurônios somam PEPS⁷ e disparam quando atingem um limiar pré-definido (*threshold*). Mas isso não é o que se observa na maioria dos neurônios.

Há características, traços operacionais, padrões de respostas, estados de excitabilidade e tipos de comportamento deveras importantes nos neurônios. A figura 2.1 ilustra alguns padrões de respostas típicos observados em neurônios naturais no córtex cerebral de mamíferos [60], [26]. Note-se a riqueza de comportamentos dos neurônios naturais. O desafio é compreender como estes padrões de disparo contribuem para a *computação* executada nos SNs, e como a interação entre neurônios resultam operações computacionais.

O conjunto de características como um todo define o processamento que a célula neural executa, e não é incomum observar que alterações de alguns parâmetros fazem uma célula neural mudar totalmente a computação que ela executa.

É comum observar um neurônio disparando regularmente e, em função do tipo de excitação ou inibição que recebe começar a disparar rajadas de pulsos. Ou ainda um neurônio permanecer silente mesmo recebendo excitação constante em uma frequência, mas disparar a partir de pequenas variações na frequência de entrada, respondendo como um ressonador (ver resumo sobre vários tipos de resposta de neurônios corticais em [26]).

2.4 Breve histórico da simulação neuronal

Quando surgiram os primeiros modelos de RNAs prevalecia a crença de que inteligência tem como base o raciocínio, cujo fundamento é a lógica. Dessa forma, em 1943, McCulloch e Pitts (M&P) criaram um modelo de *neurônio artificial* (NA) para demonstrar como as células elementares do cérebro podem computar funções lógicas elementares [22]. O modelo original de M&P usa um neurônio artificial

⁷ **Potenciais Pós-Sinápticos.** *Excitatório* (PEPS / EPSP em inglês): abalos que diminuem a diferença de potencial (tensão negativa) da membrana celular (despolarização) causados por corrente iônica que ocorre quando um *pulso* (PA) atinge a sinapse do neurônio. *Inibitório* (PIPS / IPSP): abalos que aumentam a diferença de potencial (tensão negativa) da membrana (hiperpolarização) causados por corrente iônica em virtude da chegada de um *pulso* na sinapse de um neurônio [138] pp.115-132).

simples, cujas entradas e saídas são binárias x_i (-1, 1), cada entrada está associada um peso w_i , um número real.

No modelo de M&P a cada passo computacional a saída é determinada após calcular a soma ponderada de todas as entradas do neuroide $\sum w_i x_i$; e se esta soma exceder a um limiar ($\sum w_i x_i \geq 0$), o estado do neuroide N_i é ajustado para *ativo* (1); ou em caso contrário, N_i é ajustado para *inativo* (-1) (ver [61] cap.10, [59], [8], [55]).

Todavia, neurônios não possuem um estado constantemente (ativo ou inativo). Ao contrário, eles pulsam, ora rajadas e ora disparando um simples PA e silenciando-se. E este comportamento está longe do proposto pelo primeiro modelo “tudo-ou-nada” dos neurônios artificiais de M&P. Fica claro que a finalidade dos neurônios artificiais não é mimetizar os neurônios, nem descrever matematicamente as propriedades dessas células; eles são abstrações, concebidos para executar funções matemáticas em RNAs.

Há então que se fazer uma distinção entre os modelos matemáticos de neurônios, como os desenvolvidos por Hodgkin-Huxley, discutidos adiante, e os NAs que, embora partindo de inspiração biológica, não têm compromisso com descrever ou prever processos biológicos com precisão.

Continuando a descrição dos NAs, os que sucederam aos primeiros eram similares ao modelo de M&P, como o *perceptron*, apresentado por Rosenblatt, em 1958. Modelos subsequentes de NAs começaram a usar como entradas e saídas números reais em vez de binários, e a função limiar “tudo-ou-nada” de ativação foi trocada por mapeamento entrada/saída (*adaline*) ou por funções não-lineares do tipo sigmóides [22]. Uma interpretação para uso de números reais como entrada/saída nesses NAs é que eles representam a média de disparo observada nos neurônios biológicos. Contudo, condições de computação analógica observadas em animais levaram os investigadores a revisar a representação por meio da taxa de disparo e questionar como os neurônios computam [24], assunto discutido adiante.

Todos esses modelos de neurônios artificiais permitiram o surgimento de uma série de máquinas baseadas em RNAs, como a *multilayer perceptrons*, redes de Hopfield, redes de Kohonen, máquinas de Boltzmann, etc. [59], [8], [55]. As RNAs derivadas desses modelos são poderosas ferramentas para engenharia nos domínios de

reconhecimento de padrões, controle, robótica, classificadores, dentre outras. Estas RNAs são ainda instrumento para muitas questões teóricas como calculabilidade, complexidade, capacidade, teoria da regulação, etc. (ver [22] e referências).

Maass [24] propôs uma classificação das RNAs, por conseguinte dos modelos de neurônios artificiais nelas utilizadas, em três gerações. A primeira geração é baseada nos modelos de McCulloch & Pitts (e similares) como unidades computacionais. Estes neurônios artificiais são conhecidos como *perceptrons* ou portas com limiares, conforme descrito acima.

Figura 2.2 – **Gerações de RNAs**. Maass classifica as redes neurais artificiais em três gerações [24]: a primeira, tipicamente *feed-forward*, usa *perceptron* e modelos derivados de M&P. Na segunda, tipicamente recorrentes, neurônios artificiais operam com números reais e algoritmos de correção de erros. A terceira são redes neurais pulsadas RNP.

A segunda geração é baseada em NAs que aplicam função de ativação em vez de limiares rígidos, geralmente usando funções do tipo $\alpha(y) = 1/(1+e^{-y})$, conhecidas como sigmóides. Estas redes são capazes de computar funções com entradas e saídas analógicas; portanto, podem executar ampla gama de funções matemáticas relacionando estados de entrada com os de saída. A segunda geração de RNAs também se caracteriza por formar redes neurais recorrentes (RNR) e por introduzir algoritmos de aprendizado baseados em gradientes, como retro-propagação do erro

(*error back-propagation*). RNRs podem ser correspondentes a máquinas de Turing no sentido de que podem funcionar como *aproximadores* universais de funções [9], [8].

Biologicamente, os NAs e as RNAs da segunda geração eram mais próximos da realidade. A noção de *média temporal de disparos* era bem aceita desde 1928, quando Adrian demonstrou que a taxa de disparo dos neurônios receptores nos músculos está relacionada com a força que o músculo desenvolve (ver [62], [28]).

Neurônios, especialmente em áreas corticais superiores, disparam em várias frequências entre um mínimo e um máximo; assim, ao usar números reais representando *taxas de disparos* a segunda geração apresenta, biologicamente falando, uma relação mais realística, uma interpretação mais próxima de fenômenos biológicos do que a geração anterior. A terceira geração de redes neurais artificiais é especialmente inspirada em neurônios biológicos e seus modelos são derivados de avanços recentes na neurociência [51], [22].

Na década de 1980, dados experimentais colocaram em xeque a abordagem da computação/representação por meio da taxa de disparo. Entre outros trabalhos, Perrett, Rolls e Caan [63], em 1982; depois Thorpe e colaboradores (ver [64], [62]), demonstraram que macacos e humanos executam computação de padrões visuais (e consecutiva classificação) em tempos tão curtos quanto 100-150 ms, embora o processo envolva pelo menos dez estágios sinápticos da retina até o lobo frontal. Em seguida, outros cientistas mostraram que áreas corticais envolvidas no processamento visual podem completar a computação que lhes cabe em 20-30 ms (ver [65], [66]). Mas, a taxa de disparo dos neurônios nestas computações era usualmente abaixo de 100 Hz; portanto, seriam necessários pelo menos 20-30 ms apenas para a obtenção de amostragens (*samples*) suficientes para que a *taxa de disparo* fosse calculada.

Em outras palavras, em 100-150 ms humanos e macacos respondem seletivamente a estímulos visuais complexos (como faces ou alimentos); mas devido à arquitetura *feed-forward* presente no córtex visual, composta de várias camadas de neurônios disparando a uma taxa média de 10 ms, apenas um (ou nenhum) PA poderia ter

sido disparado por cada neurônio envolvido no processo durante aquele intervalo de tempo [22], [62].

Por esse motivo, a codificação de taxas de disparo em variáveis analógicas reais parece bastante duvidosa no contexto das rápidas computações no córtex visual [24]. Outras frentes de investigações vinham mostrando por meio de dados experimentais que vários sistemas biológicos fazem uso da propagação (dos retardos) no potencial de ação isolado para codificar informações (ver [67], [24], [66] e referência aos trabalhos de Abeles, Arbib, Hopfield, Thorpe, entre outros).

Outras pesquisas no campo das atividades neurais *in vitro* e *in vivo* demonstraram que neurônios podem gerar padrões temporais de disparos com precisão de milissegundos (ver [68], [46] e referências aos trabalhos de Lindsey et al., Prut et al., Villa et al.; Chang et al.; Tetko & Villa, Mao et al.). Mais ainda, Swadlow demonstrou a importância do atraso na propagação dos PAs [69]: em córtex de coelhos, atrasos na propagação de PAs podem ser da ordem de 0,1 ms a 44 ms, dependendo do tipo de neurônio. Porém, tão importante quanto determinar a ordem de grandeza dos atrasos foi mostrar que pares de neurônios repetem propagações com precisão mais acurada que o tempo de duração dos PAs, que ocorre na ordem dos milissegundos; ou seja, neurônios repetem pulsos (a borda de subida, por exemplo) com precisão na ordem de microssegundos, enquanto PAs duram milissegundos. Por que SNs, com neurônios diferentes, mantêm atrasos de propagação com tamanha precisão se a temporização não tivesse relevância para algum tipo de computação?

Além desses, outros dados foram se somando; por exemplo, Wehr e Laurent [70] relatam a codificação temporal precisa de odores na forma de assembleias neurais em gafanhotos; enquanto Ikegaya e colegas relatam padrões de disparos espontâneos se repetindo depois de minutos tanto em culturas quanto *in vivo*, mantendo precisão de milissegundos [68].

Assim, as RNPs começaram a ganhar terreno e ganhar preferência de alguns cientistas mais interessados em simulação de fenômenos com maior proximidade aos observados em organismos biológicos. Uma revisão histórica mais completa sobre o desenvolvimento das RNAs e RNPs pode ser encontrada na tese de Timoszczuk, defendida na EPUSP ([71], caps. III e IV).

2.5 O que há na temporização dos pulsos?

Há várias possibilidades de codificação e computação por meio de PAs. Potenciais de ação, em si, não carregam informação alguma [28], e se eles fazem algum sentido para os SNs isso se deve à interação entre neurônios. Mesmo com todos os avanços na neurociência, não existe consenso sobre como os neurônios *codificam* informações; assim, existem dois grandes grupos que giram em torno de duas hipóteses principais: (1) que a codificação ocorre pela *taxa de disparo* de pulsos (*rate coding*), ou (2) que a codificação acontece na *temporização dos pulsos* (*pulse coding*) e suas variantes. O que se propõe nesta tese é uma “nova” hipótese na qual (3) a codificação e computação ocorrem por meio da *reverberação de assembleias*, ou seja, do comportamento em grupo dos neurônios. Codificação, representação, e formas de computação por assembleias são assuntos de capítulos adiante.

Considerando a codificação pela *taxa de disparo*, formas de obtenção dessa *média* definem o tipo de código ou de computação realizada nessas redes. Mais comum é obter a taxa por *contagem de disparos* durante um intervalo de tempo (*rate as a spike counting*), que é o método de se obter a média dentro de uma janela de tempo, método este equivalente a uma integração temporal. Outra forma pode ser por meio da atividade de uma população de neurônios idênticos, em redes simétricas (com o mesmo padrão de conexões sinápticas); assim, pode-se obter a taxa como uma *medida do número de neurônios* que enviam a informação a um determinado neurônio receptor (*average over several neurons*), método que pode ser visto como uma integração espacial. É possível ainda obter a taxa por meio do *cálculo da densidade de pulsos* ao longo do tempo (*rate as a spike density*), que Gerstner e Kistler explicam como uma forma de obter a média repetindo um experimento certo número de vezes (cf. [28] tópico 1.5). Há pouca probabilidade de os dois últimos métodos ocorrerem na natureza. Assim, a estratégia mais provável é que a integração temporal (contagem de pulsos) durante uma janela de tempo crie um valor representativo, como fora pensado para as RNAs de segunda geração.

Por outro lado, a hipótese da codificação e computação por *Pulse Coding* é muito mais rica e apresenta mais opções. Thorpe e colaboradores compararam alguns esquemas de codificação e decodificação aplicados a um grupo de n neurônios

pulsantes dentro de uma janela de T ms. Eles mediram a capacidade de codificação por contagem (*count code*), da codificação binária (*binary code*), da codificação por temporização (*timing code*), da codificação pelo ranking ou a ordem dos pulsos (*ranking order code*) [62], [22]; além de destacar o potencial de códigos que usam sincronismo. Possibilidades de codificação temporal são descritas a seguir, partindo da classificação de Gerstner e Kistler (cf. [28] tópico 1.6):

a) Pela latência ou tempo de atraso de um pulso (Time-to-First-Spike).

Este tipo de codificação se baseia no tempo de atraso (ou latência) dos pulsos em relação a um sinal de referência externo. Um exemplo de referência externo poderia ser, usando o exemplo de [28], o disparo para um movimento de sacada dos olhos. Assim, é possível imaginar um código no qual cada neurônio levaria em conta a temporização para a chegada do primeiro pulso a partir desse sinal de referência.

Esse tipo de codificação justifica (e soluciona) o problema proposto por Thorpe, de que um neurônio não teria tempo para disparar mais de um pulso quando processando sinais visuais rápidos (ver texto acima, [62], [22]). O tempo de atraso dos PAs e/ou o tempo de latência para o primeiro pulso pode conter todos os dados relevantes para certo conjunto de informações num tempo muito curto.

A figura 2.3 ilustra os modos de codificação de spikes: (A) ilustra a latência ou atraso de um pulso (*Time-to-First-Spike*), (B) ilustra a fase do pulso em relação a oscilações internas (*phase-shift coding*), (C) ilustra a correlação e o sincronismo (*correlations and synchrony*) e (D) ilustra a codificação pela ordem de chegada (*ranking order*).

b) Pela fase do pulso em relação a oscilações internas (Phase coding).

Se pensarmos um sinal de referência interna, por exemplo, um sinal oscilatório gerado por uma população de neurônios ou um CPG, referência esta diferente de um evento externo singular, então podemos ter um código baseado na diferença da fase entre os pulsos em codificação e este sinal periódico de referência. Investigações sobre essa classe de codificação por fase ou acoplamento de osciladores datam do meio da década de 1980.

O uso de PLLs (*phase-locked loop*) na codificação de informações também é outra linha de pesquisa dentro dessa classe de codificação. Exemplos de investigações

dessa última classe são as pesquisas desenvolvidas na EPUSP em PLLs e redes osciladoras por Piqueira e colaboradores ([15], [17], [16]).

A codificação pela fase também foi usada como hipótese por Del Nero, por exemplo, quando apresenta o modelo de controle voluntário baseado em malhas de sincronismo de fase ([2], tese EPUSP, cap. VII).

Várias linhas de pesquisas, em algum grau, adotam a hipótese de codificação pela fase do sinal, algumas usando diferentes denominações: *pulse coupled oscillators* ([72], [73], [74]), *oscillatory networks* ([75], [76]), *network synchronization* e *global synchronization* ([77], [78], [79]).

Figura 2.3 – **Codificação temporal usando spikes.** (A) A partir de um estímulo externo, pode-se codificar baseado na latência de chegada de um pulso. (B) A partir de um sinal interno cíclico, pode-se codificar usando a diferença de fase dos spikes. (C) Pode-se pensar em um código a partir de correlações internas (geralmente quando grupos de neurônios sincronizam). (D) Pode ainda haver um código baseado na ordem de chegada dos pulsos com relação a um sinal interno ou externo.

c) Pela correlação e sincronismo (Correlations and Synchrony).

Em vez de usar referências (internas ou externas) como base para codificar informação no atraso ou deslocamento de fase dos pulsos, o sistema pode codificar informações tomando como referência a relação temporal entre grupos internos de neurônios. Definimos grupos internos como aqueles ativados apenas internamente, e não por estímulos exteriores à rede.

Assim, a rede pode utilizar o sincronismo entre estes grupos como forma de retenção de dados, ou de medição de distância (ou tempo) entre atividades, ou contar a quantidade de neurônios envolvidos, entre outras formas de codificação.

A ideia mais conhecida a este respeito é que estar em sincronia pode significar “pertencer ao grupo”. Dessa forma, são pertinentes ao código os neurônios que disparam em sincronismo, ou que estão a certa “distância” (numa certa janela temporal); ou ainda, quando uma determinada quantidade de neurônios (*um padrão*) foi ativada.

Redes de neurônios podem também processar, representar, ou codificar estímulos por meio de cadeias de neurônios pulsando sincronamente, o que traz a noção de *assembleias de neurônios* [47], [80], [81], [44].

Mais abrangente que a ideia de cadeia sincronizada de neurônios (*synfire chain*) é a noção de grupos neurais pulsando “*polissincronamente*”, introduzida inicialmente por Bienenstock na forma de “*synfire braid*” [45], e posteriormente reformulada por Izhikevich na forma de *grupos policronizados* (GP ou GPs) [46], conceitos estes que serão revisitados adiante.

d) Ordem de chegada (Rank order).

A ordem de chegada dos pulsos (*ranking order*) também pode representar um sistema de codificação relevante. Thorpe e colaboradores demonstraram que uma particular ordem de chegada dos pulsos é apenas uma das $N!$ ordens possíveis, N representando o número de neurônios envolvidos na codificação [62].

Ou seja, para um grupo de 10 neurônios apenas é possível mais de 3,6 milhões de possibilidades. Um tipo de código usando *rank order* é capaz, em princípio, de transmitir até $\log_2(N!)$ bits de informação [22], [82].

e) Codificação binária (Binary code).

Outra possibilidade pode ser um grupo de neurônios usarem o código binário. Nesta codificação os neurônios tanto emitem valores binários quanto avaliam quem está recebendo uma entrada ativa (1) ou inativa (0). Várias formas de codificação podem acontecer a partir do princípio básico da codificação binária, como acontece em

lógica digital, por exemplo, que faz uso do código Gray, a codificação decimal em binário (BCD) etc.

Na forma como Thorpe e colaboradores propõem [62], o código binário aconteceria em um grupo de neurônios em paralelo, nesse caso, um grupo de N neurônios seria capaz de codificar $\log_2(2^N) = N$ bits de informação; ou seja, é igual ao próprio número de neurônios, que são capazes de realizar 2^N combinações. Contudo, a codificação binária assim proposta deve ser considerada dentro de uma janela de tempo, o que equivale a dizer que os neurônios formam uma assembleia durante este instante. Assim, esta forma de codificação é mais bem interpretada se a tratarmos como uma assembleia, e não como um código binário. O que mais difere as assembleias de um código binário é que este último não prescinde de sincronismo ou correlação temporal interna entre neurônios, porém, enquanto assembleias, sim.

Para uma análise da quantidade de bits que cada modo de codificação é capaz de desenvolver, veja [62], [82], ou [22]. Em trabalho recente, Cessac e colaboradores fazem uma revisão sobre fatos e questões envolvidas na codificação neural por PAs/*spikes*, analisando as possibilidades e restrições da codificação temporal [51].

Cabe notar que a codificação temporal pode ser dividida em três classes de acordo com as relações entre pulsos e a referência tomada, a saber. Uma das classes de códigos é obtida a partir da relação entre um sinal externo de referência e os pulsos acontecendo internamente, e.g. medindo a latência, ou a ordem de acordo com um input. Outra classe pode ser formada entre sinais de referência interno e pulsos provocados por estímulos externos ou internos, e.g. na diferença de *fase* entre este sinal base (interno) e um pulso estimulado externa ou internamente. Outro exemplo dessa classe é quando um sinal periódico interno aumenta a propensão de disparo de certos neurônios, ou seja, um sinal interno marcando um momento interno de referência para que a codificação aconteça. A terceira classe acontece pela correlação apenas interna de pulsos; mais provável entre grupos de pulsos, e.g. quando um grupo gera um padrão que serve de disparo (*triggering*) ou de base temporal para outro, como em CPGs; ou ainda quando formam cadeias síncronas (*synfire chains*) e grupos policronizados, ambos discutidos adiante.

É provável que existam outras formas de codificação envolvendo o tempo e PAs/spikes, e.g., pelo comprimento de uma rajada ou trem de pulsos (*burst*), pela ressonância de um neurônio ou um grupo (indiretamente ligando o tempo à frequência). Há também variações das hipóteses acima, e.g. as assembleias biestáveis, uma forma de correlação associada à reverberação.

Vimos assim que há diversas formas de neurônios pulsantes transformarem seus potenciais de ação em informação, e que esta informação não está nos PAs em si, mas na relação temporal entre pulsos em associação à interação que os neurônios têm uns com os outros. Os PAs são quase sempre do mesmo formato e de duração aproximadamente constante, o que nos faz descartar qualquer possibilidade de codificação em seu formato ou duração, sobrando assim como alternativa investigar as relações temporais nas interações neurais. Para isso, devemos tentar simular neurônios de sorte que reproduzam da forma mais próxima possível as respostas observadas nas células biológicas.

2.6 Reprodução do comportamento neural

A seguir é feita uma análise dos modelos de neuroides (*spiking neurons*) mais usados em simulação de RNPs, e os de maior significação histórica ou prática. A finalidade é comparar características dos modelos para investigar adequações em simulações de assembleias, ao mesmo tempo em que se avalia custo computacional do modelo, plausibilidade biológica, e outros fatores relevantes.

RELEVÂNCIA PARA A TESE:

Analisa-se a seguir alguns modelos matemáticos de neurônios artificiais pulsados (*neuroides*) a fim de se obter uma visão geral sobre quais modelos podem ser mais apropriados para uso em computação por assembleias neurais.

Um dos primeiros neuroides foram os modelos introduzidos por Hodgkin & Huxley (H&H) [83], que é um modelo realístico e biologicamente plausível em termos de descrição dos fenômenos que ocorrem na membrana neural. Trata-se de um modelo

baseado nas condutâncias da membrana celular; porém este modelo apresenta alto custo computacional.

2.6.1 Modelo de Hodgkin–Huxley

Em 1952, Hodgkin e Huxley (H&H) apresentaram um conjunto de quatro equações diferenciais ordinárias não-lineares e dez parâmetros que descrevem: o potencial de membrana da célula, a ativação das correntes de sódio (Na^+) e de potássio (K^+) e a desativação da corrente Na^+ , além de considerar um grupo de pequenas correntes de escape (*leakage current*).

Figura 2.4 – **Modelo de Hodgkin & Huxley**. (A) Circuito da membrana com condutâncias dos canais iônicos g_{Na^+} , g_{K^+} ; capacitância C , e condutâncias R das correntes de escape (*leakage currents*). (B) As equações que definem o modelo. As variáveis m , h e n descrevem a abertura e fechamento dos canais dependentes da tensão. (C) Curvas das constantes de tempo e função de ativação como $f(V)$. A interação entre estas funções resulta no potencial de ação (D), com destaque para o influxo de íons Na^+ na subida, seguido pela saída de K^+ da célula na descida do pulso.

O modelo H&H descreve como os PAs iniciam e se propagam nos axônios gigantes das células nervosas em lulas⁸. A figura 2.4 ilustra o modelo. As equações e o texto completo sobre o modelo podem ser vistos no trabalho original [83]. Uma

⁸ Por esse trabalho, a dupla foi agraciada com o prêmio Nobel, em 1963.

comparação com os primeiros modelos de neurônios pulsados podem ser encontrados em [84]. Esta tese se limita a mostrar as características do modelo.

O modelo H&H é um dos mais importantes na neurociência, e em geral os cientistas se referem a qualquer modelo de neuroide baseado em condutância como uma derivação dele, ou um modelo do *tipo H&H*. Sua importância se deve ao fato de ser capaz de descrever propriedades com significado biofísico, cujos parâmetros são mensuráveis. O modelo respeita comportamentos registrados em neurônios naturais: um incremento abrupto na subida de disparo, seguido por um curto período no qual o neurônio está inábil para disparar novamente (período refratário absoluto), seguido por um período no qual a membrana está despolarizada, o que faz com que novos disparos sejam difíceis (para períodos refratários, ver Bear et al.. [6] pp. 82-93; Kandel et al.. [3] p.157).

Além disso, o modelo permite investigar a integração sináptica, a distribuição computacional e os efeitos morfológicos na árvore dendrítica, a influência mútua em canais iônicos, além de outras questões relacionadas à dinâmica da célula. O problema é que o modelo tem alto custo computacional. Como demonstrado em [26], assumindo que cada expoente usa apenas 10 operações de ponto flutuante (FLOP), o modelo necessita 1200 FLOPs para simular apenas 1 ms de atividade de um neurônio. Ou seja, o modelo se aplica a redes com poucos neurônios, ou em casos nos quais o tempo não é um fator determinante, ou ainda quando se usa super computadores ou computação distribuída.

É possível dizer que o modelo H&H é o ponto de partida, uma referência para uma série de modelos que dele derivaram. Modelos baseados em *condutâncias* podem reproduzir os detalhes de medidas eletrofisiológicas com um alto grau de exatidão, mas devido a sua complexidade intrínseca, tais modelos são difíceis de serem analisados e apresentam alto custo computacional, ou ambos. Por esses motivos, surgiu uma classe de modelos cuja preocupação única é reproduzir o *fenômeno* da geração do PA, modelos esses chamados fenomenológicos. Estes são mais populares por serem mais fáceis de serem compreendidos e adequados para estudos de codificação neural, memória, e das dinâmicas de rede [28]. A questão que deve ser colocada é: em termos de operações computacionais realizadas nos (e

pelos) neurônios, o que se perde e o que se ganha ao escolher um modelo em detrimento de outros?

O modelo de quatro dimensões de H&H pode ser sistematicamente reduzido para duas dimensões considerando-se que a dinâmica representada pela variável m (ver fig. 2.4) é muito mais rápida que de n , h e u e isso sugere que m pode ser tratada como uma variável instantânea. Assim, faz-se uma atribuição $m(t) \leftarrow m_0[u(t)]$ que substitui a equação de m no modelo de H&H.

Por outro lado, observado o gráfico da função de ativação (fig. 2.4.C) vê-se que a curva $n_\infty(v)$ e $1-h_\infty(v)$ são praticamente idênticas. Assim, faz-se uma aproximação das variáveis n e $(1-h)$ por uma variável efetiva w (ver desenvolvimento dessas simplificações em [28], cap. 3). O que resulta são duas equações e um modelo bidimensional de neuroide:

$$F(u, w) = u - \frac{1}{3}u^3 - w$$

$$G(u, w) = b_o + b_1u - w$$

Os primeiros a proporem tal simplificação foram Fitzhugh e Nagumo, e o modelo descrito pelas duas equações acima é conhecido por modelo Fitzhugh-Nagumo [28]. O resultado é que os modelos derivados de H&H, e simplificados pelo método acima (modelos bidimensionais), conseguem reproduzir boa parte do comportamento fenomenológico dos neurônios biológicos, com uma considerável redução do custo operacional.

O modelo bidimensional pode ser visualizado e estudado no plano de fases, e o resultado é um conjunto de modelos mais fáceis de serem computados e com respostas mais fáceis de serem preditas. Nas palavras de Izhikevich, “um importante resultado dos estudos de Hodgkin e Huxley é que *neurônios são sistemas dinâmicos*, portanto eles devem ser estudados como tal” [25] p.8 (grifo do autor).

2.7 Modelos de neuroides

Grandes questões sobre simulação de RNPs estão ligadas a como representar neuroides tal que reproduzam a riqueza de respostas dos neurônios, com baixo

custo computacional, e com plausibilidade biológica. Em outras palavras, existe o problema de como escolher um modelo matemático que permita simular fenômenos mais próximos dos observados no mundo biológico e que sejam computacionalmente viáveis. Como vimos, uma das soluções foi simplificar o modelo H&H; porém, outros modelos matemáticos de neuroides surgiram e que de uma forma ou de outra permitem que sejam simuladas redes neurais pulsadas. Assim, mostraremos a seguir os modelos mais relevantes, alguns derivados de H&H e outros sem qualquer conexão com condutâncias da membrana.

As descrições a seguir são baseadas nos trabalhos de Izhikevich (“*Which Model to Use for Cortical Spiking Neurons?*” [26], “*Simple Model of Spiking Neurons*” [27]), Paugam-Moisy e Bohte (“*Computing with Spiking Neuron Networks*” [22]), Destexhe (“*Conductance-Based Integrate-and-Fire Models*” [85], “*The high-conductance state of neocortical neurons in vivo*” [84]), e Gerstner e Kistler (“*Spiking Neuron Models: Single Neurons, Populations, Plasticity*” [28]). Foram escolhidos primeiramente modelos de neurônios que podem ser expressos na forma de equações diferenciais ordinárias, depois será mostrado o modelo SRM (*spike response model*), que é um modelo que não usa equações diferenciais.

2.7.1 Modelo integra-e-dispara (I&F)

O modelo de neuroides do tipo integra-e-dispara⁹ (*integrate-and-fire*) é um dos mais conhecidos e amplamente utilizados em simulação de redes neurais pulsadas. O modelo, em uma versão muito simples, foi postulado em 1907, pelo fisiologista francês Louis Lapicque (ver [86], [87]). Lapicque propôs um circuito no qual um capacitor representa a capacitância da membrana e um resistor em paralelo drena as cargas elétricas do capacitor (*leakage resistance*).

⁹ Segue-se aqui uma nomenclatura que é uma tradução literária do termo em inglês, usada em textos científicos em português, e.g.: Pumarica, JCS (2010), “*Projetos de modelos neurais pulsados em CMOS*”, dissertação orientada por Hernandez, EPUSP; ou Simões e Costa (2007), “*Aprendizado não-supervisionado em RNPs de base radial: um estudo da capacidade de agrupamento para a classificação de Pixels*”, Rev. Controle & Automação V.18 N.2 2007 pp. 251-264.

A ideia persiste hoje nos modelos I&F, por isso, é comum encontrar denominações como *leaky integrate-and-fire (LIF)* [25], [86]. Claro, pela falta de conhecimento, Lapicque não incluiu a dinâmica de geração de *spike*, nem levantou hipóteses sobre recomposição (*reset*) da membrana depois do *spike*. O modelo I&F atual foi proposto em 1965, por Richard Stein, e o termo foi cunhado por Bruce Knight em 1972 [86].

No modelo I&F, uma corrente $I(t)$ alimenta um circuito elétrico R - C paralelo, conforme a figura 2.5. A corrente $I(t)$ se divide nas correntes I_R (resistor) e I_C (capacitor), assim a corrente total do circuito pode ser descrito pela equação:

$$I(t) = \frac{(u(t) - u_{rest})}{R} + \frac{C du}{dt}$$

Multiplicando a equação acima por R e introduzindo a constante de tempo τ_m , obtém-se a equação mais conhecida do modelo:

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t)$$

onde u é o potencial de membrana e u_{rest} é o potencial de *reset*, que geralmente é assumido como zero, e a constante de tempo $\tau_m = RC$. A equação também pode ser escrita na forma simplificada:

$$v' = I + a - bv, \quad \text{se } v \geq v_{limiar}, v \leftarrow c$$

onde v é o potencial de membrana, I é a corrente de entrada, e a , b , c e v_{limiar} são os parâmetros. Esse modelo é um integrador simples e fácil de ser implantado. O *spike* é forçado pela condição $v \geq v_{limiar}$. A iteração $v(t+1)=v(t)+(I+a-bv(t))$ utiliza só quatro operações de ponto flutuante para ser executada (4 FLOPs/ms de simulação), mais uma operação de comparação com a constante v_{limiar} ; portanto, é um modelo computacionalmente simples.

Por outro lado, é um modelo unidimensional (uma variável apenas), assim, ele não consegue reproduzir várias das respostas mostradas na fig. 2.1: não executa *phasic spiking* (B), *bursting* de qualquer natureza (e.g: C, D), *rebound responses* (M, N); além do que não permite variação de limiar, que é fixo [26].

Na verdade, o modelo só reproduz três das respostas mostradas na figura: “*tonic spiking*” (A), “*class 1 excitable*” (G), e “*integrator*” (L) [22]. Alguns pesquisadores ([25]

p.80) argumentam que este modelo nem deveria ser chamado de “*spiking neuron*” porque não possui um mecanismo de geração de *spike*, ou seja, um gerador de rampa de subida da tensão de membrana; como o fazem os modelos *quadratic I&F* (QIF) e outros modelos que derivaram do I&F, discutidos a seguir.

Figura 2.5 – **Modelo de neuroides Integra-e-dispara.** (A) Considerando dois neurônios: *i* (pré) e *j* (pós-sinápticos). (B) O gráfico ilustra como os abalos advindos do neuroide *i* (mais ruídos) modificam o potencial de membrana do neuroide *j*. Quando $u(t)$ atinge o potencial de disparo (limiar), a membrana é ajustada para o valor de *reset*. Os *spikes* de *j* foram desenhados manualmente no gráfico para fins estéticos. O *spike* não ocorre de fato neste modelo, que não tem mecanismo de produção de *spikes*, apenas a função delta é aplicada aos neurônios seguintes. (C) O modelo pode ser entendido como um circuito R-C paralelo, ao qual se associa um mecanismo de detecção da tensão da membrana que dispara a função δ quando a tensão no capacitor ultrapassa o limiar.

2.7.2 Família I&F

Como o modelo I&F é unidimensional, com uma segunda equação linear descrevendo as dinâmicas de ativação é possível aprimorar o modelo. Dessa forma, modelos de neuroides podem ser descritos como derivados do I&F formando uma família de modelos: *adaptive I&F* (aIF), *I&F-or-Burst* (bIF), *resonate-and-Fire* (rIF), ou o *quadratic I&F* (QIF) (ver [26] para síntese e análise de custos computacionais).

Desses, o modelo QIF se destaca porque pode representar várias das propriedades dinâmicas descritas na figura 2.1, como latência para disparo (*delayed spiking* - I), bi-estabilidade para repouso e disparos tônicos (*bistability* - P), atividade dependente de limiar (*threshold variability* - O). O modelo QIF é o mais simples dos neuroides LIF a capturar, qualitativamente, os comportamentos da curva $f-I$ (resposta logarítmica

da frequência de disparo perto da corrente de limiar), permitindo simular modelos mais realísticos [26]. O modelo é uma variante do *LIF* no qual $\frac{du}{dt}$ passa a depender de u^2 , mantendo-se eficiente em termos computacional – usa 7 FLOPs/ms para simular atividade de um neuroide. Por meio de uma transformação simples do potencial da membrana u para a fase θ , o neuroide *QIF* pode ser transformado em um modelo *Theta-Neuron* (canônico de Ermentrout–Kopell), não discutido em detalhes aqui. Apenas deve-se notar que se trata de um modelo unidimensional que pode ser esboçado como um diagrama de fases circular, uma proposta bastante interessante e de fácil visualização e compreensão (ver [22]).

Outro modelo importante da família é o *integra-e-dispara exponencial* com adaptação (*adaptive Exponential I&F - aEIF*), introduzido por Brette e Gerstner em 2005 [88]. Trata-se de um modelo fisiologicamente relevante no sentido de que seus parâmetros podem facilmente ser relacionados com medidas fisiológicas. O modelo combina um mecanismo exponencial de geração do *spike* com uma equação adaptativa, cujos parâmetros são extraídos por uma metodologia e protocolos eletrofisiológicos aplicada a um modelo baseado em condutância de um “*regular spiking neuron*” (similar ao *tonic spiking* (A) na fig. 2.1), que é um modelo comum no córtex. As equações que definem o modelo *aEIF* são:

$$\frac{Cdv}{dt} = f(V) - w + I$$

onde C é a capacitância da membrana, $f(V)$ é a função que caracteriza as propriedades passivas e os mecanismos de disparo do *spike*, w é uma variável de adaptação, e I é o vetor das correntes sinápticas. Os autores usaram para $f(V)$ uma combinação de funções lineares e exponenciais:

$$f(V) = -g_L(V - E_L) + g_L\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)$$

onde g_L é a condutância de fuga (*leak conductance*), E_L é o potencial de repouso, Δ_T é o fator de subida, V_T é o potencial de limiar. Depois do *spike* a equação é reiniciada com valores $V_r \leftarrow E_L$. A corrente de adaptação é definida por:

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w$$

onde τ_w é a constante de tempo, e a representa o nível de adaptação de antigos sub-limiares. A cada tempo de disparo, w é incrementada de um valor b , usado para adaptação do disparo do *spike*. Como este é um modelo de neurônio excitável, ou seja, permanece quieto e dispara quando recebe excitação, esse modelo também é do interesse para computação por assembleias. Descontando a diferença de que parâmetros realísticos (em termos biológicos) podem ser definidos por meio de dados experimentais fisiológicos neste modelo, ele é bastante parecido com o apresentado a seguir, que descreve qualitativamente os fenômenos neurais.

2.7.3 Modelo simplificado de Izhikevich

Um modelo simples de neuroide que se tornou bastante conhecido foi apresentado por [27]. Baseado na fenomenologia do comportamento de disparo dos neurônios, Izhikevich desenvolveu um modelo matemático analisando os neurônios como “sistemas dinâmicos”. Em outras palavras, o modelo difere de outros tradicionalmente baseados em “condutância” de canais iônicos no sentido de que não há neste modelo representação explícita das condutâncias e correntes iônicas fisiológicas das células, o modelo se limita a mimetizar o fenômeno de geração de *spike* e período de recuperação da membrana.

Trata-se de um modelo híbrido, que combina o melhor do comportamento excitável contíguo dos neurônios naturais, o que leva à geração de um potencial de ação com subida contínua e acentuada, com um subsequente *reset* nos estados das variáveis logo após o *spike* [89].

Por ser um modelo híbrido, o modelo consegue reproduzir, realisticamente, todos os tipos de comportamento de disparo dos neurônios biológicos mostrados na figura 2.1, sem que parâmetros do modelo tenham um significado fisiológico direto; ou seja, no modelo não existem variáveis que representem correntes ou condutâncias de íons Na^+ ou K^+ .

No modelo Izhikevich um neurônio é representado por um par de equações diferenciais simples, com baixo custo computacional:

$$C \frac{dv}{dt} = k(v - v_r)(v - v_t) - u + I$$

$$\frac{du}{dt} = a\{b(v - v_r) - u\}$$

onde C é a capacitância da membrana, v é o potencial da membrana em mV, v_r é o potencial de repouso da membrana, v_t é o potencial instantâneo do limiar, u é a variável de recuperação (ou seja, a diferença de todas as correntes de entrada e saída dos canais iônicos disparados por tensão), I é um vetor com as correntes (*inputs*) dendríticas e sinápticas (em pA); e a e b são parâmetros.

Figura 2.6 – **Modelo Simples – curvas e respostas**. No topo da figura vemos o significado de cada parâmetro do modelo: $v(t)$ é o potencial de membrana, que tão logo atinja 30 mV sofre um *reset* para c . O parâmetro a controla a taxa de decaimento do período refratário, b descreve a sensibilidade de $u(t)$ e denota o valor de *reset* de u após disparo. (B) Mapa relacionando os parâmetros às respostas mostradas no item (C), que mostra comportamentos do neuroide. Note que TC₁ e TC₂ têm regimes de disparos diferentes, dependendo do potencial de membrana em repouso. Adaptada de imagem de livre reprodução, autorização em <http://www.izhikevich.org> - acesso em 20/08/2011.

Quando a membrana atinge o pico do *spike* ($v > v_{pico}$) o modelo força o reset das seguintes variáveis: $v \leftarrow c$, e $u \leftarrow u + d$ (sendo c e d parâmetros do modelo). As duas equações diferenciais acima, adequadas para a escala de *milivolt* e *milissegundo* (com incremento do tempo discreto = 1 ms), se reduzem às seguinte equações:

$$v' = 0.04 v^2 + 5v + 140 - u + I$$

$$u' = a(bv - u)$$

com os seguintes artifícios computacionais de *reset* depois de cada *spike*:

$$\text{if } v \geq 30 \text{ mV, then } \{v \leftarrow c; u \leftarrow u+d\}$$

No modelo, $v' = dv/dt$ denota o potencial de membrana, e $u' = du/dt$ denota a variável da corrente de restabelecimento do potencial de membrana. O vetor I denota as correntes sinápticas de entrada ou uma corrente contínua injetada por estímulo externo. Este vetor pode induzir o efeito dos potenciais pós-sinápticos excitantes (PEPS) e/ou inibitórias (PIPS) advindos dos *spikes* de outros neuroides, bem como simular a influência de sinais de ruídos.

O parâmetro a descreve a escala de tempo da variável de recuperação u ; e b descreve a sensibilidade de u para flutuações no potencial de membrana v na região de tensão abaixo do limiar (*sub-threshold membrane potential*). Já c denota o valor (em mV) do potencial de restauração da membrana v imediatamente pós-disparo, enquanto o parâmetro d denota o valor de *reset* da variável u pós-disparo (*after-spike reset parameters*).

O modelo é versátil e tem poder de simular grande número de neurônios em sistemas computacionais simples (ver [90], [91]), sendo o único capaz de simular todas as respostas neurais presentes na figura 2.1; dessa forma, com um único modelo matemático de neuroide é possível simular um vasto repertório de comportamentos de disparo dos neurônios naturais, possibilitando ainda relacionar respostas de diferentes tipos de neuroides modificando alguns parâmetros. O resultado é que o modelo permite simulações complexas com baixo custo computacional, porque requer apenas 13 FLOPs para simular 1 ms de um neurônio (mais uma tomada de decisão). Não há limiar neste neurônio, pois o limiar é um valor entre -70 mV e -50 mV e é dinâmico como nos neurônios biológicos. O valor +30 mV é o valor de pico do *spike*, não um limiar em si. Todas essas vantagens justificam a escolha do modelo para simular computação por assembleias, para que em futuros estudos fique fácil simular assembleias que surgem a partir de neuroides ressonantes, por exemplo, ou por rajadas.

2.7.4 Modelo de resposta impulsiva

O modelo de resposta impulsiva¹⁰ (*Spike Response Model* - *SRM*) não compartilha com os modelos anteriores a característica de ser descrito por equações diferenciais ordinárias acopladas. Trata-se de um modelo fenomenológico focado na descrição da emissão do *spike*. O modelo foi introduzido em 1991, por Gerstner e é uma generalização do *nLIF* (*LIF* não-linear) cuja direção tomou caminhos diferentes: enquanto no *nLIF* os parâmetros são dependentes de tensão, no *SRM* eles dependem do tempo desde o último *spike*. Ainda, enquanto os modelos I&F usam equações diferenciais, o modelo *SRM* expressa o potencial de membrana do neuroide j por meio da variável $u_j(t)$ num tempo t como uma integral sobre o passado (ver [28], [22]).

Os *spikes* são gerados quando $u_j(t)$ ultrapassa um limiar v ; momento em que se define formalmente o instante de disparo do neuroide: $t_j^{(f)}$. A dinâmica do potencial de membrana é descrita por uma função $\eta(t-t_j)$, onde $(t-t_j)$ é o tempo transcorrido desde que o último limiar v_j foi excedido, ou seja, o tempo do último disparo $t_j^{(f)}$. A duração do pulso não carrega informação em si, podendo ser usada uma função δ_{abs} (impulso). O modelo prevê um período refratário absoluto, e após este intervalo o potencial de membrana pode ser expresso por:

$$\eta(s) = -v_j \exp(-(s - \delta_{abs})/\tau_{ref})$$

A fórmula simplificada para implantação do modelo é obtida assumindo que somente o último disparo t_j do neuroide j contribui para o cálculo do período refratário, e integrando a equação para uma janela de 1 ms (ver [28], [22]). A janela de 1 ms iguala a resolução temporal desse modelo aos apresentados anteriormente. Isso reduz o modelo *SRM* para a forma simplificada *SRM₀*:

$$u_j(t) = \eta_j(t - t_j) + \sum_{i \in \Gamma_j} w_{ij} \varepsilon(t - t_i - d_{ij}^{ax}), \text{ próximo } t_j^{(f)} = t \Leftrightarrow u_j(t) = v$$

Há alguns problemas com o modelo *SRM₀* no que diz respeito às respostas da figura 2.1. Quando estimulados com I constante, neurônios apresentam períodos de

¹⁰ Seguindo denominação em português usada por Timoszczuk [71].

adaptação até estabilizarem em uma frequência. Como o modelo leva em conta apenas tempo do último disparo, estes neuroides não apresentam adaptação. Ele responde bem como “*fast-spiking neuron*”, mas não como tônico ou qualquer outro com adaptação, itens (A, E, F, R) fig. 2.1. Pelo mesmo motivo, também não respondem como “*bursting neurons*”, portanto, os itens (C, D, N) na fig. 2.1 não serão alcançados com este modelo, e nem qualquer das respostas como “*rebound spikes*”, itens (M, N, S, T) [28], [22].

Em suma, este é modelo de neuroide que apresenta limitações, porém é de simples implantação e, na forma “bem-comportada” (discutido adiante) que é “*fast-spike*” ele pode ser cogitado para computação por assembleias.

2.8 Resumo do capítulo

Neste capítulo mostrou-se inicialmente que há vários níveis de investigação do funcionamento do cérebro, com o conceito de neurônio sendo central. A partir do neurônio podemos descer nas investigações da biofísica da célula neural, ou subir para as camadas superiores, nas quais se investigam circuitos e conexões neurais e a funcionalidade das redes – assunto do próximo capítulo.

Os tópicos desse capítulo destacaram como é feita a formalização do comportamento dos neurônios. Primeiro, foi mostrado um “mapa” com vários tipos de respostas obtidas de forma experimental em neurônios do córtex para servir como base para o entendimento de “o que” se quer obter quando simulamos redes neurais artificiais em computadores.

Foi mostrado que os primeiros modelos de neurônios artificiais (binários) não tinham compromisso com plausibilidade biológica e houve uma simplificação demasiada do modelo matemático [41]. Surgiu assim uma segunda geração de neurônios artificiais, que trabalha com números reais representando a “média de disparo”, tentando imitar a taxa de potenciais de ação observada nos neurônios biológicos. Essa geração de redes neurais artificiais esta mais próxima do que ocorre em termos de computação cerebral; porém, certos tipos de computação não são possíveis de serem executada utilizando o paradigma da “taxa de disparo”. Evidências mais recentes indicam que a

codificação interna usada em SNs parece estar relacionada à relação temporal dos pulsos em si e da interação entre neurônios para tratar tais sinais; assim, surge a terceira geração de redes neurais artificiais – as RNPs.

Na sequência foram discutidos os tipos de codificação possíveis quando se consideram os PAs (ou *spikes*) como unidades portadoras ou transmissoras de informação: codificação pela latência em relação a um sinal de referência, pela fase em relação a oscilações internas, pela correlação e sincronia, pela ordem de chegada dos pulsos, e pela codificação binária; além de revisar as técnicas de codificação pela taxa de disparo.

Se existem alguns modos como os neurônios podem codificar informação e realizar procedimentos computacionais por meio de spikes, quais características e propriedades do comportamento neural são importantes e devem ser reproduzidas pelos modelos matemáticos que os simulam?

Os potenciais de ação (*spikes*) em si não possuem informação, mas algumas características da geração do *spike* são importantes porque restringem a extensão operacional do neurônio. Por exemplo, os tempos de recuperação do potencial da membrana (períodos *refratários*) influenciam diretamente na capacidade do neurônio disparar de novo, e isso precisa estar reproduzido no modelo do neuroide, porque esses períodos influem na frequência máxima do neuroide, por exemplo.

Neste ponto, podemos dizer que há basicamente dois caminhos a serem tomados: é possível tentar reproduzir o comportamento e as respostas neurais por meio de equações que descrevam as respostas eletrofisiológicas da célula, ou por meio da reprodução fenomenológica da emissão dos *spikes*, analisando as células nervosas como sistemas dinâmicos. No primeiro caso, cada variável na equação do neuroide remete a um evento fisiológico observável em neurônios, enquanto no segundo caso, apenas o fenômeno do *spike* e as devidas temporizações e dinâmicas são retratadas no modelo do neuroide.

Há vários modelos matemáticos para simular neurônios artificiais pulsados (neuroides), assim, foram mostrados alguns modelos relevantes para a tese. O modelo de Hodgkin e Huxley, por motivos históricos porque dele derivam quase todos os outros. Depois, o modelo integra-e-dispara, por ser o mais usado em

simulações de RNPs. Desse modelo derivou uma família de neuroides, como: *adaptive I&F*, *I&F-or-Burst*, *resonate-and-Fire*, *quadratic I&F (QIF)*, *adaptive exponential I&F (aEIF)*.

Depois, foi discutido o modelo simples de Izhikevich que vem se popularizando pela facilidade de implantação e baixo custo computacional. Outro neuroide, o modelo de resposta impulsiva (SRM) também foi mostrado por ser um modelo que não se baseia em equações diferenciais ordinárias e é de fácil simulação.

Temos assim alguns modelos de neurônios com os quais podemos simular a computação de assembleias neurais. Qual deles escolher? Isso depende do propósito das simulações, do hardware e recursos de software disponíveis, de testes com várias simulações diferentes, de comparação de resultados entre modelos; enfim, uma resposta pode não ser tão direta quanto se desejaria.

Alguns modelos podem ser descartados em princípio, como o modelo de H&H, que apresenta precisão ao refletir a eletro-fisiologia das células, porém, com alto custo computacional. Ao menos neste primeiro momento da pesquisa, não parecem ser relevantes detalhes do comportamento fisiológico dos canais iônicos nas células, mas a plausibilidade na geração do *spike*, as temporalidades, e a dinâmica das células; por exemplo, se o modelo é capaz de disparar rajadas, se responde de imediato a um conjunto de *spikes* que gera um PEPS ou PIPS relevante.

Assim, como primeira opção para um modelo neural, escolhemos o modelo simples de Izhikevich, que será nosso modelo de referência de neuroides, com o qual pretendemos implantar experiências com coalizões. Um trabalho interessante pode ser comparar o desempenho de assembleias desse modelo com outros modelos de neuroides em várias aplicações; uma sugestão para futuras pesquisas.

O que justifica a escolha do modelo de Izhikevich é que, para um mesmo programa, será possível alterar parâmetros e obter neurônios diferentes, se comportando fenomenologicamente como integradores, ressonadores, com resposta latente, disparando rajadas; enfim, todas as respostas ilustradas na figura 2.1.

Em princípio, podemos pensar que qualquer modelo *bem-comportado* de neuroide servirá para a comprovação das hipóteses sobre computação por assembleia. Por *bem-comportado* entenda-se que são modelos de neuroides que, ao receberem um

número mínimo de *spikes* dentro de um intervalo de tempo Δt , este neuroide certamente responde disparando um *spike*, como fazem os neurônios *fast-spiking* ou *regular-spiking* (ver modelo simples, figura 2.6).

Ou seja, em um modelo bem-comportando é importante que o neuroide faça integração espacial de *spikes* e responda rápido, dentro da janela de resolução temporal do sistema de simulação.

Características adicionais (rajadas, latência, repique, etc.) podem tanto enriquecer quanto levar as assembleias a operarem de forma caótica. Certamente sistemas nervosos contam com esse tipo de comportamento dos neurônios, seja tirando proveito deles, seja tentando diminuir os efeitos colaterais que causam. Importante é que o modelo de neuroide permita facilmente produzir essas respostas, para facilitar simulações dessa natureza no futuro.

Modelos que reproduzam o fenômeno do *spike* não costumam ser problema, dada a grande quantidade disponível atualmente. Porém, há outros fenômenos associados às células biológicas que influem na computação exercida pelos neurônios, assunto do próximo capítulo.

3 PLASTICIDADES

//

Neste capítulo vamos discutir o nível intracelular e de interconexões, que pode ser considerado inferior ao nível discutido no capítulo anterior, no sentido de que neste vamos discutir processos intrínsecos, que modificam os neurônios e os ajustam de forma a permanecerem operantes dentro de certos limites. Vamos ainda analisar mecanismos de ajustes nos acoplamentos entre neurônios, e as formas como isso contribui para estabilidade operacional de toda a rede.

Sistemas nervosos e encéfalos são compostos de unidades computacionais que ajustam a si próprias, modificando pontos de operações para permanecerem ativos, assim como ajustam as sinapses de forma que o sistema não se desestabilize. Este é um feito notável que introduz um paradoxo ao sistema como um todo: SNs são “máquinas” constituídas de unidades computacionais individuais, que se ajustam com o propósito de manterem operantes tanto a si próprias quanto ao conjunto, como um todo; porém, a unidade isolada não sabe exatamente o que faz, não sabe do propósito da operação que executa, e não conhece o processo global, ou seja, não sabe da computação realizada no SN como um todo.

Dentre as propriedades dos neurônios, é possível que a mais marcante seja sua capacidade de adaptabilidade, sua plasticidade.

RELEVÂNCIA PARA A TESE:

Plasticidades descrevem como as unidades de processamento numa rede (neurônios) se modificam e modificam as conexões com outras unidades, alterando todo o comportamento e a computação realizada pela rede. Plasticidade é responsável pelo aprendizado e aquisição de novas habilidades, pela criação e manutenção de memória, pela alteração da topologia na rede. Neste tópico estão descritas as principais formas de plasticidade e como são aplicadas a RNPs.

3.1 Plasticidade celular

Acredita-se que modificações nas sinapses sejam responsáveis por toda plasticidade que ocorre no SN como um todo, especialmente no córtex. Assim, plasticidade seria responsável pela memória, aprendizado, entre outras mudanças, temporárias ou não, que ocorrem nos SNs. A exata relação entre as mudanças em

propriedades microscópicas e as consequências funcionais decorrentes dessas modificações é motivo de controvérsia. Considera-se que as propriedades plásticas das sinapses sejam o substrato biológico do aprendizado e memória. Como destacado em Morrison e colaboradores [39], de um ponto de vista teórico, as regras de modificação sináptica devem:

- (a) ter sensibilidade quanto às atividades pré e pós-sinápticas dos neurônios (hebbiana) a fim de que respondam às correlações nas entradas, que é a essência para todas as regras do *aprendizado não-supervisionado*;
- (b) ter mecanismos de seletividade às entradas, como na formação de campos receptivos na presença de traços marcantes, essência para o aprendizado durante desenvolvimento (*developmental learning*);
- (c) desenvolver alto grau de estabilidade nas memórias sinápticas e ainda se manter plásticas, essência da *formação e manutenção de memória*;
- (d) ter habilidade para levar em conta a qualidade da realização da tarefa mediada por um sinal global de sucesso, base para o aprendizado por reforço (*reinforcement learning*).

Da mesma forma como ocorre com a modelagem dos neurônios (vista no capítulo anterior) torna-se mais simples e com menor custo computacional criar modelos que descrevam o comportamento macroscópico do *fenômeno plasticidade*, em vez de detalhar a biofísica responsável por ela.

No modelo fenomenológico da plasticidade as regras de atualização das sinapses passam a depender apenas do tempo de disparo, do potencial de membrana, e do valor do peso sináptico; ou ainda, de uma versão mista que utiliza filtros passa-baixas (média temporal) aplicada às soluções anteriores.

Sinapses, como neurônios, são heterogêneas; ou seja, tipos diferentes de sinapses e neurônios possuem propriedades plásticas diferenciadas, e por isso não há um modelo único que inclua todos os tipos de plasticidade. O fato é que não existe uma regra que englobe todas as formas de plasticidade em um único modelo.

A inspiração inicial para modelos de plasticidade tem sido o postulado de Donald Hebb (de 1949), traduzido aqui a partir de [39]:

Quando um axônio de uma célula A está suficientemente perto para excitar uma célula B, ou participa de forma repetida ou persistente no disparo de B, algum processo de crescimento ou de mudança metabólica acontece em uma, ou em ambas as células, tal que a eficiência de A aumenta, como uma das células que disparam B.

Porém, o postulado de Hebb leva em conta a média de disparos de A sobre B e negligencia a relação temporal entre os disparos. Na verdade, o postulado de Hebb tem uma série de aspectos a serem questionados:

Vários aspectos importantes do postulado hebbiano não têm validade. Antes de tudo, uma ordem causal e temporal está implícita: a célula A evoca o disparo da célula B, dessa forma, a célula pré-sináptica A é ativada antes da célula pós-sináptica B [...] Segundo, a célula A precisa agir em cooperação com outras células pré-sinápticas para induzir fortalecimento sináptico [...] Terceiro, o enfraquecimento da conexão nunca é mencionado, nem está implícito [...] Quarto, o fortalecimento é sinapse específico, o que significa que somente entradas ativas são afetadas pela regra de aprendizado [...] Quinto, a plasticidade hebbiana é em si e por si própria instável: disparos correlacionados levam ao fortalecimento sináptico, que por sua vez gera mais disparos correlacionados [...] Finalmente, o postulado hebbiano trata todas as sinapses como se elas fossem iguais e é indiferente a qualquer impacto que a localização dendrítica da sinapse possa ter na indução da plasticidade sináptica, já que Hebb explicitamente trata ambas, as sinapses dendríticas e as somáticas, como sinapses somáticas [38] (tradução nossa).

Posteriormente surgiram modelos que incluem temporalidade na plasticidade sináptica, notadamente o modelo STDP (*spike-timing dependent plasticity*). Não que o modelo seja uma adaptação à regra hebbiana, ao contrário, dados experimentais mostram que a regra de Hebb é válida, e que nela pode e deve ser incluído o elemento tempo, tanto para o fortalecimento quanto para o enfraquecimento das sinapses, como será discutido adiante.

Uma revisão sobre as propriedades básicas da plasticidade sináptica numa perspectiva histórica pode ser encontrada em Sjöström e colaboradores [38]. Uma descrição cronológica de fatos marcantes na investigação sobre plasticidade pode ser encontrada em Morrison, Diesmann e Gerstner [39]. Neste mesmo texto vê-se que modelos de plasticidade baseadas em spikes e temporização podem ser feitos a partir de uma combinação dos seguintes termos:

- (a) *incremento ou decremento espontâneos* - um efeito pequeno que pode levar a um lento escalonamento homeostático dos pesos sinápticos na ausência de atividades (trata-se de uma plasticidade não-hebbiana);
- (b) *efeitos causados por spikes pós-sinápticos, independentes da chegada de spikes pré-sinápticos* - causa regulação homeostática porque, se neurônios disparam em altas taxas por horas, as sinapses serão reguladas para decrementar seus pesos (não-hebbiana).
- (c) *efeitos causados por spikes pré-independentes das variáveis pós-sinápticas* - típico de plasticidade sináptica de curto prazo (não-hebbiana);

- (d) *efeitos causados por spikes pré-sinápticos, independentes em conjunção com spikes pós-sinápticos (STDP) ou em conjunção com despolarização pós-sináptica* – que é uma regra geral hebbiana aplicada a neurônios pulsantes (é uma plasticidade hebbiana);
- (e) *todos os efeitos acima dependem do valor corrente do peso sináptico* - o resultado é que perto de um peso sináptico máximo (ou mínimo) mudanças devem se tornar cada vez menores, seguindo uma curva assintótica.

Há pelo menos três tipos de plasticidade observados em redes de neurônios biológicos que afetam diretamente o resultado da computação executada na rede. A classificação é feita levando-se em conta o tempo necessário para a *indução* das mudanças e a escala temporal da *persistência* da mudança.

O primeiro tipo é a plasticidade de curto-prazo (*short-term plasticity*), que pode ser induzida em tempos curtos, como um segundo ou menos, e persiste por alguns milissegundos. Em contrapartida, o segundo tipo, a plasticidade de longo-prazo (*long-term plasticity*), pode igualmente ser induzida em menos de um segundo ou em poucos segundos; porém, sua persistência é maior. Para se referir a esse segundo tipo são usados dois termos: LTP (*long-term potentiation*), quando a sinapse aumenta sua força de conexão, e LTD (*long-term depression*), quando há enfraquecimento da conexão sináptica [39], [28]; Esses dois termos são ainda genericamente conhecidos por STDP, e o efeito desse tipo de plasticidade é duradouro, de mais de uma hora em culturas e em preparos in vitro.

STDP oferece um poderoso meio de conformar e modificar topologias em circuitos neurais, embora seja improvável que essa plasticidade atue sozinha. Em organismos no estágio pós-natal há evidência de desenvolvimento de características básicas que desafiam a estabilidade dos circuitos, por exemplo, ocorrem mudanças muito rápidas nas entradas sensoriais, de tal forma que STDP não pode explicar sozinha tais transformações [92]. Tais considerações têm levado à pesquisa de mecanismos de plasticidade que atuem sinérgica ou complementarmente à STDP, e que sirvam para restringir força sináptica e intensidade de disparo nos neurônios.

Adicional à plasticidade de curto-prazo e à STDP há um terceiro tipo, que é denominada plasticidade *homeostática* (PH), ou intrínseca. Na PH operam um conjunto de mecanismos que: (1) atuam no controle da força sináptica total do neurônio, modulando sua excitabilidade intrínseca em função da média de atividade

do neurônio, (2) criam a habilidade nas sinapses de submeterem modificações hebbianas à dependência de sua história [92].

Nos SNs, neurônios devem manter-se operacionais, ou seja, não podem sofrer potenciação ou depressão a ponto de se tornarem super-excitáveis ou demasiado insensíveis a estímulos. Em outras palavras, independente da LTP ou LTD que tenham sofrido, neurônios devem ter um mecanismo de regulação global das sinapses, incrementando ou diminuindo sua excitabilidade como um todo, a fim de convergir para um ponto operacionalmente estável, o que ocorre numa escala de tempo de minutos ou horas [32], [34], [33].

Há inúmeros trabalhos sobre mecanismos de plasticidade em geral, destacam-se aqui as revisões em [33], [39], [92] e [36]. O texto a seguir é baseado ainda nos trabalhos: [32], [34], [93], [94], [95], [96], [97].

A modelagem fenomenológica da plasticidade neural tem ganhado atenção na simulação de RNPs em geral, e são de especial interesse na simulação de coalizões. Dessa forma, na medida em que forem introduzidos os conceitos serão discutidas suas implantações em RNPs, além de discutir sua importância para computação por assembleias.

3.2 Plasticidades de curto-prazo (*short-term plasticity*)

A função primária dos neurônios é receber sinais eletroquímicos de outros neurônios, executar alguma operação sobre eles (e.g. integrá-los) e retransmitir informações na forma de sinais elétricos ou químicos para outros neurônios. Em resposta a sucessivos estímulos, neurônios podem mudar tanto a força de suas conexões quanto parâmetros operacionais internos, tornando-se mais ou menos sensíveis a futuros estímulos. Esse efeito pode ser imediato, num curto intervalo de tempo, ou mais prolongado.

Sinapses biológicas possuem dinâmicas que controlam a amplitude da resposta pós-sináptica em função do padrão temporal dos estímulos pré-sinápticos, especialmente quando recebem trens de pulsos (rajadas). Ou seja, ao receber uma rajada de pulsos, cada *spike* sucessivo consegue evocar uma resposta (um PEPS

ou PIPS) menor (depressão), ou maior (facilitação) que o *spike* anterior. Isso ocorre numa escala de tempo de 100 ms até 1 segundo.

Os mecanismos pelos quais esse fenômeno acontece permanecem em discussão; porém, parecem estar relacionados com distribuição *quantal*¹¹ de neurotransmissores nas sinapses. Este tipo de fenômeno foi abordado por Markram e colaboradores [37], que apresentou um modelo fenomenológico para simulação desse tipo de adaptação sináptica rápida [98]. Uma representação gráfica do fenômeno pode ser visto na figura seguinte.

Figura 3.1 – **Adaptação sináptica de curto-prazo**. Gráficos mostrando dados experimentais a partir de Markram et al. [98], obtidos de neurônios do córtex de ratos. (A) Após um tempo sem receber *spikes* o neurônio *j* recompõe sua sinapse, e ao receber o primeiro *spike* o potencial evocado volta a ser intenso (*facilitação*). (B) A amplitude média dos potenciais evocados no neurônio *j* (pós-sináptico) decresce a cada *spike* sucessivo advindo de uma rajada do neurônio *i* (*depressão*).

Num estado de repouso, ou seja, algum tempo sem receber spikes, a eficácia da sinapse será máxima, talvez porque toda quantidade de neurotransmissor estará disponível na sinapse (fig. 3.1A). Por outro lado, ao receber uma rajada de PAs, a eficiência da sinapse em gerar potenciais pós-sinápticos diminui sucessivamente após cada *spike* recebido (fig. 3.1B). Após algum tempo sem receber PAs a sinapse retorna ao valor de repouso. Esse mecanismo de plasticidade faz sentido quando se observa que, ao receber estímulos constantes, os neurônios devem diminuir sua

¹¹ Sobre liberação *quantal* de neurotransmissores, ver: Kandel, Schwartz e Jessel, [3] p. 258.

sensibilidade, ou sua capacidade para gerar PEPS, caso contrário poderiam facilmente sair de um ponto operacionalmente estável.

Um dos modelos mais conhecido e usado para simular o fenômeno é o de Markram [98], que pode ser definido pelas equações:

$$\frac{dR}{dt} = \frac{1 - R}{D} - R w \delta(t - t_n)$$

$$\frac{dw}{dt} = \frac{(U - w)}{F} + U(1 - w) \delta(t - t_n)$$

onde R é a variável que representa a *depressão* e w a variável que representa a *facilitação*; U , F , e D são parâmetros obtidos experimentalmente; $\delta(t - t_n)$ é um *spike* de um neurônio i ocorrendo no tempo t_n , calculado pela função delta Dirac. O produto $R w \delta(t - t_n)$ representa a fração de neurotransmissores disponíveis no tempo t . Assim, a cada disparo do neurônio pré-sináptico i no tempo t_n a variável R é decrementada por $R w$, enquanto w é incrementada de $U(1 - w)$.

Algumas simulações em redes neurais pulsadas têm inserido algum tipo de plasticidade de curto-prazo; por exemplo, em [90] usa-se o modelo desenvolvido por Markram e colaboradores, descrito acima; mas existem ainda os modelos de Abbott, de Tsodyks-Markram (ver [39]), além de outros. A questão que se coloca é como este tipo de plasticidade pode afetar computação por assembleias?

A plasticidade de curto prazo não permite que um neurônio saia de um estado responsivo quando recebe rajadas de pulsos (*bursts*), nem quando recebe estímulos contínuos em alta frequência. A forma como estamos propondo a operação das assembleias não envolve, nesse momento, o efeito de *bursts* ativando grupos de neurônios.

Contudo, quando grupos de neurônios muito próximos formam coalizões, quase nenhum atraso de propagação ocorre entre os grupos, o que indica que as assembleias podem reverberar em alta frequência. Dessa forma, a inclusão desse tipo de plasticidade deve trazer algum tipo de estabilidade para as coalizões neurais nesta situação. A influência dessa plasticidade nas operações computacionais das assembleias deverá ser analisada ainda com mais detalhe.

3.3 Plasticidade hebbiana (*long-term plasticity*)

Modificações no acoplamento sináptico entre neurônios é um mecanismo poderoso para modificar a topologia, a computação, certas propriedades de resposta dos neurônios e toda a resposta da rede; contudo, é também um mecanismo perigoso: a menos que se controle o grau de acoplamento das sinapses, as conexões dos circuitos podem crescer ou diminuir incontrolavelmente, podendo levar a rede a atividades fora do controle. A regra hebbiana de correção do peso sináptico é intuitiva, porém, sempre houve dificuldade de explicar como ela acontece de fato.

Há algum tempo era do conhecimento científico que atividade pré- (em i) precedendo um disparo pós-sináptico (despolarização em j) induz potenciação de longo-prazo (LTP); embora não houvesse evidências de que a relação contrária causasse depressão de longo prazo (LTD); ou seja, não era certo que atividade pós-, antes da chegada de um disparo no pré-sináptico, causasse efeito contrário à potenciação. Em meados da década de 1990, evidências mostraram como a relação temporal entre *spikes* pré e pós-sinápticos pode induzir tanto LTP quanto LTD. A teoria e o mecanismo desenvolvidos a partir desse conjunto de dados são conhecidos como STDP (*spike-timing dependent plasticity*).

STDP pode ser encontrada em neurônios de espécies bastante diferentes: gafanhotos (*Schistocerca americana*), peixe-elétrico, tentilhões, girinos da espécie *Xenopus laevis*, ratos, e provavelmente em humanos. A preservação da STDP por vários milhões de anos de evolução sugere que a assimetria específica dessa regra de aprendizado é crucial para o funcionamento apropriado dos sistemas nervosos [38]. A figura 3.2 ilustra o conceito.

O modelo mais aplicado na implantação da regra STDP consiste em parear o tempo de disparo pós- em relação a um disparo pré-sináptico. Tomando o pulso pré- como referência (linha central na fig. 3.2A), as sinapses w_{ij} são ajustadas por um valor que decai exponencialmente à medida que o tempo entre os dois disparos torna-se grande $|\Delta t| = (t_j^f - t_i^f) \gg 0$.

O tempo de disparo pós-sináptico é denotado por t_j^f e o pré- é denotado por t_i^f . A sinapse w_{ij} é modificada pela seguinte regra: se o disparo de j ocorre depois de i

($\Delta t > 0$) é porque há grande probabilidade do disparo pré- ter influenciado o pós-. Quanto menor o intervalo entre $i \rightarrow j$, maior deve ter sido a contribuição do *spike* pré- para o pós-sináptico: o peso sináptico é então incrementado (potenciação).

Figura 3.2 – **STDP: Spike-timing dependent plasticity**. (A) A linha central no gráfico indica t_0 no qual ocorre um disparo pré-sináptico i . As curvas LTP (F_+) e LTD (F_-) mostram a relação entre um disparo do neurônio pós-sináptico j em relação à t_0 . Se j dispara após a chegada de um *spike* pré (i) ($t > 0$), é grande a chance do *spike* pré- ter provocado o pós (i); assim a sinapse é incrementada (potenciação) conforme a curva (F_+), que decresce exponencialmente à medida que j ocorre muito depois de i . Por outro lado, se um disparo pós (j) ocorre antes da chegada do pré (i) ($t < 0$, ou seja, j ocorre antes de t_0), significa que o pulso i em nada contribuiu para o j ; assim, a sinapse é decrementada (depressão) conforme a curva (F_-), exponencialmente. (B) O gráfico mostra *spikes* de i e j no tempo, a curva $x_i(t)$ mostra a curva de atuação pré (i) enquanto $y_j(t)$ mostra a curva de atuação de j (pós). A atualização do peso sináptico w_{ij} ocorre no momento do disparo pós (*potenciação*: pré-então-pós) proporcional a $x_i(t)$, e também no momento do disparo pré (*depreciação*: pós-então-pré) proporcional a $y_j(t)$. Figuras adaptadas de [90] e [39].

Por outro lado, quando o disparo j ocorre antes de i , ou seja, ($\Delta t < 0$), significa que o *spike* pós- não teve influência do *spike* pré-sináptico; assim, o peso sináptico é decrescido (ocorre depressão). Uma regra genérica de atualização pode ser:

$$\Delta w^+ = F_+(w) \cdot e^{\left(-\frac{|\Delta t_+|}{\tau_+}\right)}; \quad \text{sendo } |\Delta t| > 0$$

$$\Delta w^- = -F_-(w) \cdot e^{\left(-\frac{|\Delta t_-|}{\tau_-}\right)}; \quad \text{sendo } |\Delta t| > 0$$

onde $\Delta t_+ = (t_j^f - t_i^f)$ denota a diferença entre o disparo do neurônio pós-sináptico j e o *spike* pré-sináptico i , e $\Delta t_- = (t_i^f - t_j^f)$ denota a diferença entre o disparo de i para o disparo de j ; enquanto $F_{\pm}(w)$ descreve a dependência da atualização do peso da sinapse em análise. Portanto, para execução do algoritmo é preciso definir a forma de $F_{\pm}(w)$ e calcular todos os pares de sinapses que estão sujeitos à regra. Apesar de biologicamente haver neurônios que apresentam as curvas de potenciação e

depressão diferentes [39], [36], o uso de uma regra comum já, como as fórmulas mostradas na figura 3.2A produzem bons resultados em simulações computacionais.

Uma forma comumente usada para aplicar o algoritmo é escolher cada neurônio pós (j) que dispara em um tempo t_j^f e avaliar todas as suas entradas para todos os *spikes* pré-sinápticos t_i^f acontecidos numa janela de 20 ms. Para implantar a regra, seguindo [39], podem-se utilizar duas variáveis, uma $x_i(t)$ que é incrementada de uma unidade toda vez que o neurônio pré-sináptico i dispara, decrescendo exponencialmente de acordo com a constante de tempo τ_x :

$$\frac{dx_i}{dt} = -\frac{x_i}{\tau_x} + \sum_{t_i^f} \delta(t - t_i^f)$$

Similarmente, a variável $y_j(t)$ é incrementada de uma unidade toda vez que o neurônio pós-sináptico j dispara, decrescendo exponencialmente de acordo com a constante de tempo τ_y :

$$\frac{dy_j}{dt} = -\frac{y_j}{\tau_y} + \sum_{t_j^f} \delta(t - t_j^f)$$

Um gráfico da regra aplicado às duas variáveis pode ser visto na figura 3.2B. Quando ocorre um disparo pré-, decrementa-se o peso w_{ij} proporcional ao valor instantâneo da variável $y_j(t)$. Por outro lado, na ocorrência de um *spike* pós- o peso w_{ij} é incrementado proporcionalmente ao valor instantâneo da variável $x_i(t)$.

É necessário frisar que modelos de STDP provocam indução da potenciação ou depressão das sinapses, mas não as mantém. Sinapses biológicas tomam 60 minutos ou mais para consolidar as mudanças. Acredita-se que o processo de consolidação dependa de mecanismos moleculares diferentes daqueles que provocam a indução da STDP [39], [36]; ou seja, é provável que a consolidação dependa de transcrições genéticas para criar proteínas novas que incrementem a eficiência sináptica. Um artifício computacional simples de manutenção pode ser implantado adicionando sobre o STDP uma dinâmica lenta e biestável:

$$\frac{dw}{dt} = \alpha(w) = -w(1-w)(w_\theta - w)/\tau_\alpha$$

Onde τ_α é uma constante de tempo de consolidação numa faixa de vários minutos ou horas. Na ausência de estímulo, sinapses individualmente evoluem para algum dos *atratores* estáveis binários 0 ou 1, que são pontos de dinâmica lenta. Isso resulta que, depois de algum tempo, são necessários estímulos bastante fortes para causar qualquer perturbação na dinâmica sináptica [39].

Como STDP pode influir na computação por assembleias? No atual estágio da pesquisa, as simulações de RNPs com assembleias reverberantes se baseiam em redes determinísticas, não exigindo nenhuma regra hebbiana de aprendizado. Neste momento, o algoritmo STDP será introduzido em assembleias principalmente para simulação de funções lógicas básicas. Fica como proposta para futuros estudos desenvolver algoritmos nos quais assembleias se formem espontaneamente, ou seguindo alguma estrutura e organização interna pré-definida. Nesse sentido, na medida em que algoritmos de aprendizado forem pensados para as assembleias, o uso de STDP será imprescindível. O uso de STDP é também imprescindível para implantação do mecanismo de aprendizado temporal, discutidos adiante nos tópicos 3.6 (Plasticidade estrutural) e 3.7.1 (Aprendizado temporal), nesta tese.

3.4 Plasticidade homeostática

Em neurociência, plasticidade homeostática, ou plasticidade intrínseca, é um termo que se refere à capacidade dos neurônios de se auto-regularem, ou regularem sua excitabilidade em decorrência do regime de atividade a que está submetido. Trata-se de um mecanismo compensatório para ajustar a excitabilidade neural.

É bom deixar claro a diferença desse mecanismo com a plasticidade de curto-prazo (item 3.2). Na *short-term plasticity*, neurônios que recebem *spikes* em alta frequência diminuem progressivamente os PEPS a cada *spike*, para que isso não torne o neurônio inoperante em curto prazo. Como define Turrigiano [93],

a homeostatic form of plasticity is one that acts to stabilize the activity of a neuron or neuronal circuit in the face of perturbations, such as changes in cell size or in synapse number or strength, that alter excitability.

Em alguns textos usa-se o termo *escalonamento sináptico homeostático* para esta plasticidade, em virtude de sugerir uma mudança de escala global dos pesos

sinápticos na célula. Explica-se: a forma mais estudada de adaptação dependente da atividade é a mudança da plasticidade das sinapses na forma hebbiana, que, como dissemos, apresenta uma série de aspectos deficientes para explicar como ocorre a dinâmica de mudança de pesos sinápticos e dendríticos (ver na introdução do capítulo a citação de Sjöström e colaboradores [38]).

Grosso modo, a plasticidade hebbiana propõe que as sinapses se modificam de forma rápida e associativa. Quanto mais uma sinapse for utilizada mais ela é modificada, provocando potenciação (LTP). Hebb nunca sugeriu a existência do inverso de seu postulado: a contrapartida da LTP, ou seja, a depreciação (LTD) ou enfraquecimento das sinapses [38].

Assim, a plasticidade hebbiana é um processo de *feedback positivo*. Num processo de LTP as sinapses podem ser modificadas no sentido de reforçar o peso da conexão e se tornam mais excitáveis, aumentando a sensibilidade e a probabilidade de disparar novamente, reduzindo o limiar de forma a provocar novo LTP. Isso só aumenta a propensão para atingir extremos de excitabilidade [33], [93].

Hebbian (or associative) forms of plasticity have a particularly strong destabilizing effect on network activity, because they are essentially positive feedback rules that tend to drive synaptic strengths towards their maximum or minimum values [32].

A fim de evitar estados operacionais extremos, mecanismos de *feedback negativo* de auto-regulação entram em atividade nos neurônios com a finalidade de restringir níveis de atividades operacionais:

[several experiments] lend support to the notion that cortical and hippocampal pyramidal neurons have a target firing rate, and synaptic strengths are regulated to maintain these rates relatively constant in the face of perturbations in input [93].

Ou seja, evidências indicam que neurônios são células programadas para disparar dentro de uma faixa operacional. Dessa forma, os mecanismos de *feedback negativo* levam a adaptação lenta em torno de regiões operacionalmente ativas. O conceito da plasticidade homeostática é ilustrado na figura 3.3.

Os meios pelos quais as células realizam tais regulações ainda estão sob investigação e são motivos para constantes debates. Sabe-se que modificações ocorrem em mecanismos pré- e pós-sinápticos, no corpo das células, no número de

emissores e receptores de neurotransmissores; e as investigações continuam procurando evidências se tais mudanças são mais locais (nas sinapses) ou globais (na célula toda).

Figura 3.3 – **Plasticidade homeostática**. (A) Turrigiano [93] propõe uma forma robusta para entender como o neurônio ajusta o limiar (um *atrator*) para taxa de disparo ideal: observando a interseção de duas curvas que descrevem forças opostas e regulam a *acumulação de receptores*. As curvas, um *fator de escala sináptico*, retratam vários mecanismos celulares, sem designar um atributo particular. O fator Fs^+ aumenta sua capacidade de ativação quando as atividades crescem, e este fator escala para baixo a força das sinapses excitatórias. Diminuindo a atividade neural a capacidade de Fs^+ diminui; com consequente aumento da acumulação de receptores e aumento do fator de escala das sinapses. A curva Fs^- mostra uma força oposta: ela também aumenta sua capacidade de ativação quando há aumento da atividade neural, porém, com o aumento de atividade ela escala para cima a força sináptica, vice-versa se diminuir a atividade. O neurônio obtém a taxa de disparo (taxa alvo) na equalização destas duas forças. Modificações na magnitude dos fatores mudam o perfil das curvas. Se uma das curvas se modifica, e.g. a redução na magnitude de Fs^+ como na linha tracejada, o ponto de atividade se desloca (figura adaptada de [93]). (B) Mecanismos hebbianos (e.g. STDP) tendem a levar a atividade sináptica para os extremos: disparando no máximo ou silenciando neurônios. Uma das modalidades de *PH* leva o neurônio a disparar na faixa operacionalmente viável por meio do incremento ou decremento de valores sinápticos (figura adaptada de [34]).

Evidências sugerem que neurônios detectam mudanças em seus regimes de disparo por meio de sensores dependentes de Ca^{++} que regulam o incremento ou decremento de receptores de glutamato nos sítios sinápticos [93].

Funções adicionais podem usar caminhos paralelos, permitindo que mudanças locais ou globais (inclusive em toda a rede) ocorram por meio de mecanismos que operam em escalas temporais e espaciais diferentes. Foge ao escopo desta tese aprofundar sobre tais mecanismos, podendo o leitor recorrer aos textos apontados acima [33], [38], [93]. Na dificuldade de definição dos mecanismos, o procedimento mais usual é criar modelagens do fenômeno baseadas em experimentação.

Um dos primeiros modelos a considerar modificações sinápticas em função da média de atividade neural foi o proposto por Bienenstock, Cooper e Munro, que sugeriram o mecanismo conhecido por modelo BCM (ver [99]). No modelo BCM atividades pré- e pós-sinápticas são correlacionadas de forma que evocam LTP quando as *taxas de disparo* pós-sinápticas são maiores do que um limiar, e evocam LTD quando estão abaixo desse limiar. A ideia é um artifício computacional atrativo; por exemplo, o limiar aumenta para neuroides pós-sinápticos bastante ativos, fazendo com que LTP sejam mais difíceis e LTD sejam induzidas mais facilmente. Contudo, evidências em neurônios biológicos indicam que um limiar deslizante é uma noção bastante indireta [36].

Um dos modelos mais simples para implantação de PH é o apresentado por Triesch [100]. O autor modela plasticidade intrínseca por:

$$y = g_{ab}(x) = \frac{1}{1 + \exp(-(ax + b))}$$

onde x denota o total da corrente sináptica chegando ao corpo, e y denota a taxa de disparo do neurônio. Os parâmetros a e b mudam a reta e a compensação da não-linearidade do sigmoide. A regra de atualização é direta: $a \leftarrow a + \Delta_a$ e $b \leftarrow b + \Delta_b$, sendo Δ_a e Δ_b calculado por:

$$\Delta_a = \eta_{IP} \left(\frac{1}{a} + x - \left(2 + \frac{1}{\mu} \right) xy + \frac{1}{\mu} xy^2 \right)$$

$$\Delta_b = \eta_{IP} \left(1 - \left(2 + \frac{1}{\mu} \right) y + \frac{1}{\mu} y^2 \right)$$

em que μ é a média de atividade desejada para um neurônio, e η_{IP} é uma pequena taxa de aprendizado. Segundo o autor, esta regra é muito similar a uma derivada por Bell e Sejnowski (1995 – não incluído na literatura) para um neurônio com sigmoide maximizar sua entropia [100]; a regra é estritamente local, e as únicas quantidades usadas para atualizar a função de transferência não linear do neuroide são x e y . Além disso, a regra se baseia somente na atividade corrente do neuroide (y) e no quadrado dessa atividade (y^2), o que pode manter um correlato estimado com a concentração de cálcio interno da célula do neurônio.

Em [100], o autor relata uma série de experimentos realizados com o modelo, como testar o comportamento para diferentes distribuições da corrente total sináptica, resposta à súbita privação de estímulos, além de propor uma regra para interação entre PH e as regras de aprendizado hebbiano, discutidas abaixo.

O resultado é que um mecanismo como a PH, numa escala de tempo longa, tende a incrementar a ação de todas as entradas sinápticas num neuroide há muito inativo. Isso implica que sinais de ruídos inerentes ao sistema provoquem eventuais disparos em neuroides inativos, mantendo-os sempre em um limite minimamente operacional. Por outro lado, o mecanismo tende a decrementar a importância dos sinais de entrada em neuroides que disparam demasiado, para que não fujam de certos limites operacionais e desequilibrem a rede.

Interessa a este trabalho o fato de que o escalonamento sináptico, promovido pela PH, ajusta o acoplamento (força sináptica) dos neurônios com sinapses excitatórias, aumentando ou diminuindo o fator de conexão dessas sinapses, a fim de estabilizar os níveis de atividades dos neurônios [93]. Devemos questionar então o que isso representa para uma assembleia biestável, nas quais os neuroides reverberam entre si. Em que sentido a PH pode influenciar nas coalizões?

Na formação de coalizões o papel desse mecanismo de plasticidade intrínseca pode ser determinante. Um neuroide que há tempos não dispara pode se tornar mais suscetível à excitação e começar a participar de assembleias, numa espécie de “inclusão social”. Uma vez disponível e com maior grau de excitabilidade, um neuroide provavelmente entrará em formações de assembleias com vizinhos.

A plasticidade garante que eles estejam em condições de serem membros ativos e participarem de coalizões, melhorando a distribuição topográfica das formações de assembleias. Por outro lado, se uma coalizão reverbera em alta frequência a tendência é que o mecanismo cause um arrefecimento da assembleia ou da cadeia de assembleias, o que pode ser bom para estabilizar as coalizões evitando ressonância global de neuroides. Em outra frente, se uma coalizão não se forma há muito, a tendência é que os neuroides cada vez menos consigam formá-la porque os pesos sinápticos tendem a se enfraquecer, assim, esses neuroides ficam “livres” para serem cooptados e se ajustam fazendo outras coalizões; portanto, este seria

um mecanismo de “apagar” certas memórias ativadas por assembleias biestáveis, conforme se discute adiante.

3.5 Unindo as regras de plasticidade

Embora tenham ocorrido progressos nas investigações sobre interações entre os diversos mecanismos de plasticidade, muito a respeito desses tópicos permanece amplamente inexplorado; por exemplo, é necessário investigar como STDP influencia na plasticidade homeostática e vice-versa, especialmente em ambiente com sinapses muito heterogêneas, ou ainda como se relacionam e se influem quando acontecem simultaneamente, e como isso influencia na computação geral da rede. De acordo com Morrison e colaboradores [39], o maior desafio na área é executar estudos analíticos e simulações que identifiquem e caracterizem seus efeitos opostos, ou antagônicos, e investigar suas consequências funcionais.

Alguns progressos tem acontecido na área, por exemplo, Toyoizumi e colegas [101] mostraram um modelo no qual apresentam uma hipótese de que a dinâmica sináptica é controlada por três princípios: (1) sinapses adaptam seus pesos para que os neurônios efetivamente transmitam informações; (2) processos homeostáticos estabilizam a média de disparo dos neurônios pós-sinápticos; e (3) sinapses fracas se adaptam mais lentamente do que as fortes e, além disso, manter as sinapses fortes tem alto custo energético. O modelo é baseado na Teoria da Informação, e o cerne da proposta consiste na expressão:

$$\mathcal{L} = I - \gamma D - \lambda \Psi$$

onde $\mathcal{L}$ denota um critério de otimização, um gradiente de atualização de pesos sinápticos otimizado na direção ascendente; o I é a informação a ser maximizada, o termo D compara a atual distribuição de *spikes* de saída $P(Y)$ com a que seria uma distribuição ideal gerada pelo mesmo neurônio em uma taxa alvo (ou seja, plasticidade homeostática - na prática, D mede a diferença entre a taxa atual de disparo e uma taxa alvo), γ e λ são fatores de controle da importância relativa dos três termos diferentes, e finalmente, Ψ é um termo associado ao custo energético da manutenção da sinapse (ou seja, ter sinapses fortes diminui o objetivo da função,

além de aumentar o gasto fisiológico de energia para mantê-las; portanto, devem ser minimizadas).

Em outras palavras, o que a relação acima descreve é que as sinapses ajustam seus pesos tal que estejam aptas para transmitir *informação* eficientemente, mantendo uma média de taxa de disparo e seus pesos sinápticos em valores baixos (ver detalhes do desenvolvimento do modelo em [101]). Os autores descrevem que a "regra de atualização sináptica", criada por eles e que compartilha propriedades com STDP, é sensível a correlações nas entradas e é útil para memória sináptica.

Em outros trabalhos os autores utilizam regras simples para implantação da plasticidade de curto termo, como Izhikevich e Edelman [91]:

$$\frac{dx}{dt} = \frac{(1-x)}{\tau_x}; \quad \text{se spike pré-}, \text{então } x \leftarrow p_x$$

onde τ_x é uma constante de tempo e p_x é um valor de *reset*. A variável x é um fator escalar que multiplica o peso sináptico, e tende sempre a recuperar seu equilíbrio em $x=1$, dependendo da constante de tempo, porém é reiniciada a cada *spike* pré-sináptico, para o valor p_x . O parâmetro $p<1$ resulta em depressão sináptica de curto-prazo enquanto $p>1$ resulta facilitação de curto-prazo. No mesmo trabalho, os autores utilizam um mecanismo de atualização de STDP, mas não há referências a modelo de manutenção das LTP ou LTD, nem a mecanismos de plasticidade homeostática.

Em outro trabalho de simulação neural em grande escala [90], Izhikevich e Edelman utilizam o modelo de Markram para simular plasticidade sináptica de curto prazo. A dinâmica sináptica consiste em dois componentes: um passivo em que as mudanças são independentes das atividades da rede, e o outro ativo e dependente da atividade (STDP). A dinâmica passiva muda o peso sináptico lentamente, incrementando o peso sináptico mesmo que o neurônio pós-sináptico j esteja quiescente. Diferente da regra STDP que considera todos os pares possíveis de *spikes* pré- e pós-, eles assumem que só os pares vizinhos próximos contribuem para a plasticidade: se j dispara antes, a sinapse é decrescida de $F \cdot e^{(t_j-t)/\tau_-}$; e se i dispara antes, a sinapse é acrescida de $F \cdot e^{(t_i-t)/\tau_+}$; onde τ_i é o tempo de chegada do último *spike* do i ésimo

neurônio pré-sináptico; ou seja, o efeito do último *spike* se sobrepõe aos efeitos dos anteriores.

Em resumo, os autores utilizam uma regra de atualização fixa que sempre colocam os neurônios num certo ponto operacional, enquanto apostam na aplicação da STDP para ajustar os valores sinápticos. Claro, a questão endereçada no referido trabalho era estudar como STDP pode levar à auto-organização em um modelo no qual as sinapses são consideradas populações [90], e não o estudo direto das interações de mecanismos de plasticidade. Ainda assim, é possível inferir que a regra usada por eles, que ceifa o valor máximo dos pesos sinápticos em 0,5, explica a ausência do mecanismo de plasticidade homeostática: a dinâmica passiva força sempre os neurônios para um estado operacional incrementando os pesos, a STDP ajusta os pesos das sinapses pela atividade, e uma regra força um valor máximo, que podemos entender como um regulador *homeostático* do neuroide.

Outra pesquisa realizada sobre a interação de plasticidade intrínseca (PH) com aprendizado foi levada a cabo por Triesch [100]. O autor parte de uma regra hebbiana básica:

$$\Delta \mathbf{w} = \eta_H \mathbf{u} \mathbf{y}(\mathbf{u}) = \eta_H \mathbf{u} g_{ab}(\mathbf{w}^T \mathbf{u})$$

onde $\mathbf{u}$ é o vetor das entradas sinápticas, e η_H é uma pequena taxa de aprendizado hebbiano. Note-se que, sendo $\mathbf{y}$ positivo, $\Delta \mathbf{w}_i$ sempre será positivo se $\mathbf{u}_i$ for positivo, ou seja, o peso sináptico $\mathbf{w}_i$ sempre aumentará. Para evitar isso, é usual fazer uma normalização do peso $\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\|$. O autor estudou influências da plasticidade intrínseca sobre a regra simples; e.g. quando o fator de aprendizado da plasticidade intrínseca (η_{PI}) é muito maior que a do aprendizado hebbiano $\eta_{PI} \gg \eta_H$, ou seja, quando a PH é rápida. Estudou também o uso de regras hebbianas alternativas, e ainda relacionou PH com as regras, ou o modelo BCM (ver [100]).

Estes são alguns exemplos do que tem sido feito no sentido de juntar os mecanismos de plasticidade em modelos únicos, e de esforços investigativos no sentido de descrever as possíveis interações entre modelos e como elas refletem na computação e no comportamento da rede. Como apontam Morrison e colaboradores [39], “o principal desafio é realizar estudos analíticos e simulações que possam identificar e caracterizar seus efeitos compostos, e investigar suas consequências

funcionais.” Estudar os efeitos compostos desses mecanismos em assembleias biestáveis e computação por assembleias será um desafio próximo.

3.6 Plasticidade estrutural: reconfigurando conexões

A plasticidade cerebral, mais precisamente, do córtex cerebral, envolve uma combinação de mudanças funcionais (nos neurônios) e estruturais (nas conexões) entre eles [102]. Mudanças funcionais causam fortalecimento ou enfraquecimento das conexões sinápticas (e.g. STDP), sendo muito relevantes na plasticidade cerebral. Até pouco tempo acreditava-se ser esta a única responsável por mudanças na configuração da rede neuronal. Por outro lado, plasticidade estrutural está envolvida em remodelagens que levam à reconfiguração das ligações físicas nos circuitos neurais, ou seja, com a criação e eliminação de sinapses.

RELEVÂNCIA PARA A TESE:

Pensava-se que mudar o peso sináptico fosse “a” plasticidade responsável pela flexibilidade em redes neurais, capaz de explicar as adaptações funcionais nos cérebros. Porém, estudos revelam que no córtex as conexões não são tão fixas quanto em outras regiões dos SNs, e que modificações nas conexões neurais são intensas e importantes para a plasticidade cortical. Essa modificação é chamada plasticidade estrutural. Esse conhecimento tem relevância para expandir a ideia das assembleias para sistemas maduros, por exemplo, com características cognitivas, em estudos futuros. Este mecanismo é tão importante quanto as plasticidades hebbiana, homeostática e de curto-prazo.

Esse tipo de “reconfiguração” dos circuitos neurais acontece por meio de modificações das posições físicas de certos locais na árvore dendrítica, genericamente denominados *espinhos dendríticos*¹², nos quais ocorre a maior parte das sinapses de todo o córtex. Espinhos são sítios especializados, locais com grande densidade de canais receptores de neurotransmissores, com grande propensão à conexão sináptica. Esses sítios podem ser do tipo passante (ao longo do dendrito – *boutons ou shaft synapses*) ou terminal (uma saliência – *spines ou*

¹² Seguindo denominação genérica usada no livro “*Neurociências: Desvendando o Sistema Nervoso - 3a. Ed.*”, [6]. Outros textos diferenciam os espinhos (*spine synapse*) e áreas ativas no corpo do dendrito (*shaft synapses*) [3].

terminal synapses). De acordo com Kandel e colaboradores, frequentemente estes sítios fazem conexões excitatórias [3] p. 226.

Consequentemente se acontecem modificações nas posições físicas dos espinhos, elas se refletem nas sinapses associadas aos espinhos, modificando as conexões do circuito neural como um todo. Em outras palavras, a plasticidade estrutural está relacionada com mudanças de posição física das conexões neurais, e isso tem reflexos na estrutura, na topologia, e na temporização da rede neural.

Synaptic plasticity in adult neural circuits may involve the strengthening or weakening of existing synapses as well as structural plasticity, including synapse formation and elimination. [...] Although the overall cell-specific morphology of axons and dendrites, as well as of a subpopulation of small synaptic structures, are remarkably stable, there is increasing evidence that experience-dependent plasticity of specific circuits in the somatosensory and visual cortex involves cell type-specific structural plasticity: some boutons and dendritic spines appear and disappear, accompanied by synapse formation and elimination, respectively [103].

Trabalhos recentes têm identificado a remodelagem axonal, o crescimento de novos espinhos dendríticos, e a renovação sináptica (rotatividade - *turnover*) como um conjunto de mecanismos de modificações estruturais dependentes de experiência importantes em córtex maduros. Resultados de pesquisas na área começaram a revelar como ocorrem as *reconexões* (*rewiring*) nos circuitos do córtex adulto, trazendo novos *insights* sobre mecanismos de aprendizado e memória [102].

Com o avanço tecnológico na instrumentação, avanços genéticos na criação de ratos transgênicos para experimentos *in vivo*, e novos métodos de investigação, pesquisadores têm acrescentado dados e hipóteses à plasticidade cortical, em específico, e à plasticidade cerebral, no geral (revisão em [29]). Numa descrição global do fenômeno, podemos resumir: “Changes in synaptic connectivity patterns through the formation and elimination of dendritic spines may contribute to structural plasticity in the brain” [104]. O que se sabe sobre a forma como acontecem tais modificações será brevemente discutido a seguir.

O leitor interessado encontrará mais sobre o assunto em: “*Cortical rewiring and information storage*”, Chklovskii et al. [30]; “*Geometry and Structural Plasticity of Synaptic Connectivity*”, Stepanyants et al. [104]; “*Cell Type-Specific Structural Plasticity of Axonal Branches and Boutons in the Adult Neocórtex*”, De Paola et al. [31]; “*Anatomical and Physiological Plasticity of Dendritic Spines*”, Alvarez & Sabatini

[29]; “*Experience-dependent structural synaptic plasticity in the mammalian brain*”, Holtmaat & Svoboda [103] ; “*Cortical rewiring and information storage*”, Barnes & Finnerty [102]. Devemos observar que não há um termo estabelecido para denominar essa plasticidade; é possível encontrar: *plasticidades* geométrica, estrutural, anatômica, morfológica, *reconexão* cortical, etc.

Grosso modo, em neurônios excitatórios a maior parte das sinapses glutamatérgicas é feita no topo das cabeças dos espinhos dendríticos, cada qual hospedando o terminal pós-sináptico de uma única sinapse. Estudos demonstram *in vivo* que os espinhos são estruturas móveis e plásticas cuja morfologia e expectativa de vida são influenciadas pelas entradas sensoriais, mesmo em animais adultos. Em contrapartida, a maioria dos espinhos que surgem em animais adultos é transiente, assim, são raros os casos de adição de espinhos estáveis e de neles se formarem novas sinapses.

Estudos *in vitro* demonstram que padrões de atividades, que sabidamente induzem plasticidade funcional, podem causar mudanças na morfologia da espinha. Dessa forma, como destacam Alvarez e Sabatini [29], é tentador especular que mudanças plásticas da morfologia dos espinhos refletem o estado dinâmico das sinapses a eles associadas, e que tais mudanças são responsáveis, em certo grau, pelo remodelamento da circuitaria neural. Todavia, mudanças morfológicas não são obrigatórias para todas as formas de plasticidade sináptica, e ainda não está claro se mudanças na forma e tamanho dos espinhos dendríticos promovem algum impacto nos sinais sinápticos (PEPS ou PIPS).

Espinhas surgem e desaparecem durante toda a vida do animal, apenas sua taxa de reposição (*turnover*) declina com a idade. A taxa de eliminação de espinhas em ratos pós-natal é maior do que na vida adulta. A estabilidade espinhal observada em ratos adultos é, portanto, reflexo da estabilização do número de espinhas e suas localizações, o que ocorre por meio de uma progressiva e lenta queda na taxa de eliminação de espinhas em função da idade (ver [29]).

Depois do período pós-natal, a taxa de formação de espinhas mantém-se constante durante a maior parte da vida; dessa forma, com a maturidade do SN a estabilidade dos circuitos neurais é alcançada por meio do decréscimo na taxa de eliminação,

pelo incremento na meia-vida dos espinhos, e, em adição à formação e eliminação de espinhos, as *protusões* dendríticas mudam: de forma, volume, comprimento e largura. Estas mudanças no formato das conexões podem refletir ou serem a causa de mudanças funcionais nas sinapses.

Rearranjos morfológicos nos espinhos podem ser induzidos *in vitro* por estímulos. Indução de LTP é acompanhada por alargamento da cabeça do espinho, e a indução de LTD acompanha uma redução do tamanho do espinho. Assim, o tamanho e formato do espinho são regulados dinamicamente pelo padrão de atividade das sinapses associadas [29].

Em todas as idades do animal, é a atividade neural que regula a reposição das sinapses. Por exemplo, novos espinhos que persistem por vários dias têm alta probabilidade de formarem sinapses. Portanto, o efeito em longo prazo dessa plasticidade, que promove um rearranjo estrutural, é uma mudança na conectividade neural, ou seja, uma reconfiguração topológica da rede.

Em suma, existe uma dinâmica de criação e desaparecimentos de sítios nos dendritos que facilitam a criação de sinapses em locais diferentes da árvore. Durante toda a vida do animal existe uma constante modificação das posições das sinapses no córtex. A taxa de eliminação desses pontos de conexão é mais alta após o nascimento. Nos SNs adultos, ocorre certa estabilidade entre a criação e eliminação desses espinhos dendríticos; porém, a mobilidade desses pontos de conexão sináptica ocorre durante toda a vida do cérebro e pode ser responsável por grande parte das mudanças estruturais que ocorrem no córtex. Isso significa que as sinapses mudam de posição na árvore dendrítica dos neurônios com frequência.

Por outro lado, a posição da sinapse na árvore dendrítica é importante para determinar o tipo de computação que ocorre no neurônio antes mesmo de o *spike* chegar ao corpo do neurônio, discutido no tópico seguinte.

Outro fator importante é que, se há modificação na posição das sinapses, esta plasticidade abre espaço para indagações sobre novos mecanismos de aprendizado e memória [102]. É possível que representações de mundo, memórias, e operações computacionais se estabeleçam ou desapareçam em função dessas mudanças de posições das sinapses na árvore dendrítica.

Como discutido adiante, abre ainda espaço para a hipótese sobre ajustes temporais entre neurônios; ou seja, mudanças nas distâncias das conexões sinápticas (físicas ou virtuais) podem significar mudanças em atrasos de propagação dos *spikes*, o que pode ser de veras importante para a formação de assembleias.

Dessa forma, esta constante migração de posições das conexões sinápticas, ainda que menos dinâmica em SNs maduros, é real e parece ser mais que um fenômeno bizarro que acontece em tecidos corticais, e merece ser investigada com um pouco mais de atenção. Para isso, temos que entender o que ocorre em termos de operações computacionais na árvore dendrítica.

3.7 Computação dendrítica

A visão preponderante na atual neurociência tem sido que o processamento de informação nas redes neurais é resultante das propriedades das sinapses e da conectividade dos neurônios ou da topologia dos circuitos. Nesse panorama, a *excitabilidade intrínseca*¹³ dos neurônios tem um papel menor. Como consequência, a contribuição de um neurônio isolado para a computação cerebral tem sido em muito subestimada [41].

RELEVÂNCIA PARA A TESE:

Discute-se aqui em que sentido as operações computacionais na árvore dendrítica podem influenciar formação de coalizões, e como a computação nos dendritos pode ser implantada nos modelos de RNPs. A árvore dendrítica, em si, é um local onde ocorrem várias operações computacionais relevantes para as assembleias.

A questão colocada por vários cientistas é que dendritos exibem uma série de mecanismos, lineares e não-lineares, que permite que neles ocorram operações computacionais elementares, anteriores até a operações ocorridas no corpo celular:

¹³ Alguns dendritos são ativos, ou seja, são capazes de gerar potenciais de ação locais quando excitados sobremaneira [88], [68]. Por exemplo: “muitos dendritos certamente não são passivos e possuem um significativo número de canais de Na^+ , Ca^{++} e K^+ dependentes de tensão, usualmente com um número de canais insuficiente para gerar PAs completos como nos axônios. Entretanto, agem como amplificadores de pequenos PEPs gerados nos dendritos que, caso contrário, diminuiriam em até desaparecer em um dendrito (passivo) longo (excerto modificado de [2] p.126).

amplificações, execuções de funções lógicas, detecção de coincidência etc., como será discutido adiante neste tópico.

O resumo a seguir está baseado principalmente nos textos, revisões e livros: “*Biophysics of Computation*”, Koch, [7]; “*Dendritic Computation*”, London & Häusser, [41]; “*Dendritic Processing*” Segev & London, em “*The Handbook of Brain Theory and Neural Networks 2nd Ed.*” pp. 324-332, Arbib editor, [4]; “*Pyramidal neurons: dendritic structure and synaptic integration*”, Spruston [42]; “*Dendritic Excitability and Synaptic Plasticity*”, Sjöström et al. [38].

Em comum os textos destacam um assunto que emergiu na metade da década de 1990, quando novas técnicas permitiram investigar melhor os dendritos e as sinapses, introduzindo uma nova suspeita: dendritos devem executar operações computacionais muito além da que lhes são atribuídas. Nas palavras de Segev e London ([4] p. 324), texto de 1995:

For the first time ever, systematic and intimate electrical and optical visits became possible to the dendrites - the largest component of the mammalian brain in both surface area and volume. These ongoing visits have yielded a much more fascinating picture of the electrical behavior and chemical properties of dendrites than one could have imagined only a few years ago.

Até então, pensava-se que os dendritos fossem apenas passivos, distribuídos e arborizados para facilitar a conexão com outros neurônios, e assim eram modelados por equações do cabo, representadas por componentes passivos, que emprestavam à modelagem uma característica dissipativa e atenuadora dos sinais.

Mas, áreas diferentes dos encéfalos são especializadas em funções computacionais específicas e, em cada uma delas é possível observar diferentes estruturas dendríticas, o que leva a inferir que propriedades morfológicas e elétricas dos dendritos, em algumas áreas do cérebro, têm um papel específico na execução de certas tarefas computacionais. Infelizmente, na maior parte das vezes, é difícil definir a função computacional exata que um neurônio executa numa certa região; e com raras exceções, é difícil conhecer a natureza da sinapse que um neurônio particular recebe enquanto o sistema como um todo executa sua tarefa [4].

Por isso, os cientistas têm focado em explorar o *fenômeno*, ou seja, procuram classificar e explicar tipos de computação que uma célula pode executar já em suas árvores dendríticas e em suas entradas sinápticas.

Até o momento foram identificadas algumas formas de computação que podem ser descritas como: (a) seletividade pela direção de movimento, (b) detecção de coincidência, (c) extração de traços, (d) classificação de entradas, (e) operações lógicas, (f) aprendizado dendrítico (revisão em [41]; sumário em [4] p. 331).

Figura 3.4 – **Computação dendrítica**. Quatro mecanismos-chave de computação dendrítica destacados num neurônio piramidal: (A) interações entre sinapses excitatórias e inibitórias em pequenos ramos dendríticos podem executar operações lógicas. O nó s (no destaque) soma as correntes de dois sub-ramos executando a função **OU**. Cada um dos sub-ramos fornece corrente para o nó s , que logicamente significa uma função (**OU**), somente se acontecerem excitações (**E**), e (**NÃO**) acontecerem inibições, denotado por $s = (a \cdot \bar{b}) + (c \cdot \bar{d})$. (B) Em neurônios piramidais do córtex, a detecção de coincidência nos compartimentos apicais e basais é obtida por meio da retro-propagação (*backpropagation*) de um potencial de ação que coincide com uma entrada sináptica distal, o que dispara um *spike* Ca⁺⁺, que por sua vez despolariza toda a árvore dendrítica apical e impulsiona uma rajada de disparos ao axônio. (C) Dendritos passivos agem como filtros porque ao longo da árvore os dendritos atenuam altas frequências, agindo como filtro passa-baixas. (D) Dendritos podem reduzir ou amplificar a interação mútua entre entradas excitatórias. Em neurônios com dendritos ativos, grupos (ou, um cacho) de dendritos ativos sincronamente (círculos cheios) num mesmo cacho pode evocar um *spike* local (no dendrito), o que leva a uma significativa amplificação do sinal de entrada. Contudo, sinapses em um ramo separado (fora do cacho - círculos abertos) são apenas levemente influenciadas (adaptada de London & Häusser [41] p. 508). (E) Ilustração sobre como é computada a árvore dendrítica em compartimentos.

Exemplos de computação nos dendritos podem ser vistos na figura 3.4. Além dessas, sabe-se que existe *excitabilidade* nas árvores dendríticas; ou seja, alguns dendritos são *ativos*, capazes de gerar potenciais de ação locais, importantes na indução das plasticidades sinápticas [42], [41], [6] p.126.

Sjöström e colaboradores [105] defendem que a regulação da *excitabilidade dependente de atividade* nos dendritos significa que a plasticidade sináptica deve controlar indiretamente a computação dendrítica, da mesma forma que a integração dendrítica controla a plasticidade sináptica, criando uma intrincada interdependência: (a) sinapses transmitem informação ao soma via dendritos para dispararem PAs no axônio, (b) dendritos ajudam a determinar a plasticidade sináptica e (c) em seguida a atividade sináptica regula a excitabilidade dendrítica.

Os autores argumentam que deve existir um loop recíproco dependente da atividade entre plasticidade sináptica e excitabilidade dendrítica. Assim, infere-se que os dendritos são capazes de computar, além de retro-propagar PAs, que ao final é uma das formas como ocorre sinalização de tempos de disparo pré- versus pós-sinápticos que leva a cabo a regra da plasticidade STDP nas células [105].

Essa noção de retro-propagação dos PAs traz à tona uma importante questão, referente à direção da propagação dos PAs nas árvores dendríticas. Pensava-se que os potenciais de ação apenas se propagam do axônio e direção aos neurônios seguintes, mas não é o caso, como apontado por Vetter e colaboradores:

The propagation of action potentials in the dendritic tree has been the subject of great interest for over a century, beginning with the speculations of Ramón y Cajal (1904) on the directional flow of signals in neurites. During the past few decades, evidence has accumulated to suggest that under most conditions, the sodium action potential (AP), the output signal of the neuron, is initiated in the axon and retrogradely invades the dendritic tree, a process known as backpropagation [43].

A localização das sinapses na árvore dendrítica, e o tipo de excitabilidade dendrítica associada à sinapse, têm papéis decisivos na determinação das propriedades plásticas da sinapse. Mais que isso, uma vez que as propriedades elétricas da árvore não são estáticas e podem ser alteradas por moduladores (neurotransmissores) e pela própria atividade sináptica, regras de aprendizado podem ser dinamicamente formatadas pelo ajuste da função dendrítica. Esta relação recíproca entre plasticidade da *excitabilidade dendrítica* e *plasticidade sináptica* tem mudado a visão sobre o processamento de informação e armazenamento na memória em redes neuronais (revisão em [38]). Em outras palavras, não ocorre apenas plasticidade nas sinapses, mas a própria árvore dendrítica é plástica em mais de um aspecto, e deve haver um laço recíproco entre essas plasticidades.

É sabido que árvores dendríticas de vários neurônios, especialmente dos neurônios corticais, são cobertas de sítios de conexão com vários formatos, e que axônios podem fazer conexões em mais de um desses sítios (ver tópico anterior). Também é sabido que ocorrem atrasos na propagação de *spikes* durante o percurso na árvore dendrítica. Contudo, é complicado calcular tais atrasos uma vez que o diâmetro e a ramificação nos dendritos são variáveis em toda extensão da árvore. Detalhes sobre como determinar a constante de tempo média τ_m da membrana num dendrito, assim como os problemas correlatos, podem ser encontrados em Koch [7] pp.75-84.

Vetter e colaboradores simularam e analisaram a influência (e eficiência) da morfologia na propagação de PAs gerados internamente na própria árvore (*forward propagation*), além dos efeitos da retro-propagação (*backpropagation*) de PAs gerados no axônio sobre o corpo dendrítico. Os autores destacam que existem diferenças sistemáticas entre os tipos de neurônios, tanto na retro-propagação quanto na propagação direta, mesmo em condições experimentais similares, e apontam que a morfologia dendrítica tem papel importante na determinação da propagação direta ou na retro-propagação dos PAs nos dendritos [43]. A amplitude da questão da propagação na árvore dendrítica está destacada no texto abaixo:

The time course of the synaptic conductance change may vary by one or two orders of magnitude. The fast excitatory (AMPA) and inhibitory (GABAA) inputs operate on a time scale of a few milliseconds and have a peak conductance on the order of 1 nS; this peak conductance is approximately ten times larger than the slow excitatory (NMDA) and inhibitory (GABAB) inputs, which both act on a time scale of 10–100 ms [4] pp. 332-335.

Exatamente por haver variedade de neurônios e nos formatos de árvores dendríticas, e por apresentarem várias relações de retro-propagação e propagação direta, além de várias velocidades de condução na árvore (ou vários retardos de propagação), é que se prefere modelar o *fenômeno do atraso de propagação* nas árvores dendríticas de forma mais pragmática, a partir de dados experimentais, e estipular um valor entre zero e alguns milissegundos, discutido na sequência.

Em suma, neurônios possuem árvores dendríticas diferentes em diferentes regiões do cérebro, e estas podem ser passivas ou ativas, com capacidade de realizar várias operações computacionais em função do local onde as sinapses acontecem. Além disso, as posições das sinapses se alteram em função da plasticidade estrutural e a árvore dendrítica influi no atraso da propagação direta e na retro-propagação. Isso

implica que há neurônios nos quais as regras de plasticidade sinápticas (STDP) não se aplicam à totalidade de sua árvore dendrítica. Exemplos são fornecidos por Vetter et al. [43]: células Purkinje sofrem menos STDP do que células com arborização dendrítica menor, como as piramidais no córtex. Mais que isso, as árvores dendríticas são plásticas, modificando-se tanto na fase de desenvolvimento quanto na fase adulta, como visto acima. Parece evidente que todas essas propriedades influem sobremaneira no resultado da computação executada em um neurônio.

Assim sendo, como simular as operações computacionais que ocorrem nos dendritos e em que sentido isso é importante para assembleias?

Há ferramentas apropriadas para simulação de computação na árvore dendrítica, como os programas NEURON e GENESIS (ver revisão sobre simulação neural e ferramentas em [23]). Uma das soluções para simulação em ambientes menos especializados é dividir a árvore em compartimentos, como mostrado na figura 3.4E. Como explica Izhikevich ([25], pp. 292-294), cada compartimento pode ser implantado por meio de um neuroide conectado a outro via uma sinapse elétrica (*gap-junction*), formando assim os compartimentos e as ramificações da árvore dendrítica.

A abordagem da computação por assembleias considera que os grupos de neurônios já executam funções básicas, e.g. as booleanas, na própria ação de formarem coalizões. Neste estágio da pesquisa, considerar operações similares na própria árvore dendrítica seria descer um nível na investigação: as coalizões e a computação por assembleias são uma investigação no nível do “comportamento social dos neuroides”, enquanto uma investigação no nível da computação intracelular de cada unidade seria bastante complexo, inviável para este momento.

Dessa forma, assume-se aqui que uma possível computação no nível dos dendritos de uma unidade (as operações ilustradas na figura 3.4) pode estar representada nas simulações das assembleias por mais de um neurônio, agindo como uma das ramificações de uma unidade maior, um grupo. Uma assembleia, grosso modo, poderá então ser vista como uma grande unidade de processamento. Por isso, está fora do escopo desta tese detalhar uma abordagem de computação de múltiplos compartimentos.

Dois fatores cruciais na computação por assembleias são os *pesos sinápticos* e os *atrasos nas propagações*, e as árvores dendríticas têm influência direta nesses dois fatores, razão pela qual este tópico foi inserido na tese. Não vamos abordar o processamento por compartimentos na árvore, mas o processamento dendrítico influencia diretamente tanto os pesos sinápticos quanto os atrasos nas propagações.

Com relação às modificações dos pesos sinápticos, pelo menos neste estágio da pesquisa, assume-se que a rede é formada por neuroides menos arborizados (como os neurônios do córtex), assim, vamos aplicar as regras de plasticidade de curto-prazo, STDP, e plasticidade homeostática discutidas anteriormente como se a morfologia da árvore não influenciasse nessas plasticidades. Como vimos, Vetter e colaboradores [43] mostraram que células muito arborizadas não sofrem estas plasticidades de forma uniforme. Dessa forma, ao pensar assembleias em circuitos com células arborizadas (como as células Purkinje) será assumido que estas assembleias não sofrem as mesmas regras de plasticidade celular e sináptica; ou seja, assembleias em circuitos com neuroides muito arborizados são mais fixas, menos plásticas e mais *hardwired*. Isso, para manter certa plausibilidade biológica; porque se o intuito for apenas criar um aparato computacional, nada impede a utilização das regras.

A abordagem das assembleias defendida nesta tese considera que neurônios executam funções lógicas ao se agruparem temporalmente. O segundo fator crucial para formação de assembleias pode então ser favorecido pela plasticidade e/ou pela computação dendrítica. Em outras palavras, pode haver uma flexibilização nas relações temporais entre neurônios justamente porque a árvore dendrítica é plástica.

3.7.1 Aprendizado temporal

Como já foi dito, um componente importante na abordagem da computação por assembleias, além dos pesos sinápticos, é o retardo na propagação dos *spikes*, apesar da visão prevalente em RNAs seja que só pesos sinápticos são modificados, como neste trecho: “Em todos os modelos de neurônios, a maioria dos parâmetros são constantes e específicos para cada neurônio. As exceções são as conexões sinápticas que são as bases da adaptação e aprendizado [...]” [22] p.17.

A visão desta tese é diferente. Uma hipótese é que os tempos podem ser ajustados para que neurônios / neuroides sejam cooptados para novas coalizões. Assim sendo, como seria possível modificar as relações temporais entre neuroides? Como a modificação dos retardos de propagação de *spikes* tem importância para assembleias, representação, memória e controle computacional?

Modificar a relação temporal entre neuroides pode significar para um neuroide se capacitar para participar de novas assembleias. Coalizões se formam baseadas em coincidência temporal, discutido adiante. Sair de uma assembleia é simples: pode ser obtido apenas tornando nulo o peso sináptico de um neuroide j em relação aos outros neuroides $i_1...i_n$ (pré-sinápticos) que excitam a coalizão. Mas entrar em uma nova assembleia pode não depender apenas de potencializar uma conexão sináptica, porque um pulso chegando isolado pode significar nada para o neurônio pós-sináptico j . É preciso que algum ajuste temporal aconteça de sorte que o neuroide j receba *spikes* dentro de uma janela de tempo precisa, uma vez que o próprio neurônio j pode ser um intermediário numa cadeia para excitar um neurônio k em outra assembleia.

Uma das principais características das coalizões é que elas acontecem num tempo efêmero, ou seja, uma janela de tempo muito curta. Assim, algum mecanismo de ajuste fino nos tempos de propagação deve ocorrer para i ser importante na excitação de j . A hipótese considerada nesta tese é que *esse tipo de ajuste pode ocorrer nas árvores dendríticas*, em função da plasticidade nelas observadas, como foi discutido anteriormente. Mas, de que forma a computação e plasticidade na árvore dendrítica pode modificar a relação temporal entre os neurônios? Em que se baseia a hipótese da adaptação temporal nos dendritos?

Podemos ter três componentes nos atrasos de propagação de *spikes* entre neurônios: (1) atrasos nos axônios, (2) nas sinapses químicas, e (3) no retardo que ocorre na árvore dendrítica.

Foi dito anteriormente que um dos fatores a gerar os atrasos é a distância entre os neurônios, uma vez que a propagação do sinal através do axônio causa um retardo. Portanto, o primeiro componente na modelagem dos atrasos será assumido como sendo o retardo que advém do tamanho do axônio.

Some-se a este o tempo da propagação do sinal pela própria fenda sináptica. Kandel e colaboradores explicam que nas conexões elétricas (*gap-junctions*) o tempo de atraso introduzido pelas sinapses é desprezível. Por outro lado, as sinapses químicas podem gerar atrasos consideráveis – entre 0,3 e 5 milissegundos, com valor típico de 1 ms (Kandel, [3] p.176 e cap.10).

Apesar de introduzirem atrasos, as sinapses químicas amplificam sinais, por isso são vantajosas. A questão é que essa amplificação advém da ação dos neurotransmissores, que nesta fase da pesquisa não estão sendo considerados na modelagem da rede. Assim, vamos assumir um atraso médio e constante para as transmissões sinápticas de 1ms (quando aplicável), e assumir que o modelo de adequação temporal da árvore dendrítica responda por eventuais modificações ocorridas nos atrasos sinápticos químicos. Em outras palavras, os atrasos nas sinapses ou serão constantes no modelo, ou serão adequados juntos com as regras (no mesmo algoritmo) para modificação de atrasos dendríticos.

Depois que o potencial pós-sináptico é gerado (PEPS ou PIPS) existe um tempo de propagação pela árvore dendrítica. Como vimos, a árvore dendrítica é plástica, dessa forma, é de se esperar que modificações na relação temporal entre neurônios ocorram em função dessa plasticidade estrutural. Se somados os efeitos de atrasos sinápticos e da plasticidade dendríticas em um único modelo, o fenômeno no atraso pós-axônio poderá ser computado por um único algoritmo.

Em resumo, é possível que exista uma *plasticidade temporal* pós-axônio, plasticidade esta que ocorre por dois mecanismos principais: (1) a modulação de neurotransmissores que modificam o tempo de propagação das sinapses químicas em até 5ms; e a plasticidade dendrítica, que pode tanto modificar sua excitabilidade quanto pode sofrer alteração na posição física da sinapse, resultando alterações temporais que podem variar até 10 ms.

Dessa forma, para a modelagem do fenômeno de ajuste no atraso entre conexões sinápticas, um algoritmo que modifique o tempo de propagação dos *spikes* entre 0-10 ms (usando média de retardo *sináptico* mais *dendrítico* com alguma folga), pode ser suficiente para representar a dinâmica de ajuste temporal. A esse intervalo (0-10 ms) soma-se então o tempo de atraso no axônio (determinístico, fixo no modelo da

rede, representando a distância entre neurônios). A soma desses dois componentes representa assim os atrasos de até 44 ms na propagação de *spikes* experimentalmente observados em neurônios do córtex em mamíferos [46]. Este valor de 44 ms não é um máximo com o qual devemos nos preocupar, é um exemplo de um dado experimental.

Como modelar esta regra de plasticidade temporal em redes de assembleias? Uma das formas de ajuste nos tempos que pode servir para auto-organização na formação de coalizões pode se basear na regra proposta por Natschläger e Ruf [106]. Os autores desenvolveram um algoritmo de aprendizado sem-supervisão, em um trabalho cujo propósito era realizar computação equivalente às das RNA tradicionais usando RNPs: o algoritmo favorece o agrupamento de neurônios tendo os tempos dos *spikes* como entrada. A figura 3.5 ilustra como funciona o algoritmo.

Figura 3.5 – **Aprendizado temporal.** (A) Um dos métodos de aprendizado temporal consiste em utilizar múltiplas conexões entre um neuroide x_i e outro y_j , cada conexão (SC_k) com um peso sináptico w_{ij}^k associado a um atraso axonal (d^k) específicos. Ou seja, entre dois neuroides x_i e y_j há um atraso axonal fixo (DA), em série com k caminhos de $1 \dots k$ ms de atraso, cada um com o peso w_{ij}^k relativo àquela conexão. Na medida em que a rede passa por experiências, a regra STDP altera o peso de uma das conexões entre o neuroide x_i e outro y_j , de sorte que aquela que tiver o atraso mais apropriado terá o peso sináptico mais forte. (B) O gráfico mostra como um *spike* de x_1 se propaga por 10 conexões em paralelo, cada uma com atraso $\delta t = 1$ ms de para outra, cujos pesos w_{ij}^k são mostrados à esquerda no gráfico. Na primeira conexão (SC_1) o pulso sofre atenuação para 0,2 do pico do *spike* e um atraso de 1 ms. Na segunda, sofre atenuação para 0,5 e atraso de 2 ms, e assim sucessivamente. O pulso que causa maior PEPS é o da sexta conexão (SC_6) para ($w_{ij}^6 = 0,9$) com um atraso de 6 ms em relação ao *spike* original (figura e gráfico adaptados de [106]).

Numa rede de duas camadas de neuroides, i sendo o pré- e j sendo o pós-sináptico, uma conexão individual entre duas unidades é representada (computada) por um

conjunto de m de sinapses em paralelo, cada qual com um peso w_{ji}^k e um atraso de propagação d^k . O atraso d^k de um dado terminal sináptico k é definido como sendo a diferença entre o tempo de disparo do neuroide pré- i e o tempo do *spike* no pós-sináptico j . Para as m conexões sinápticas em paralelo, os autores consideram um atraso incremental entre as conexões: $d^{k+1}=d^k+\delta t$.

No modelo original, Natschläger e Ruf usam uma regra de aprendizado “vencedor-leva-tudo” [22], [106]; na qual se um PEPS em uma conexão sináptica k precede imediatamente um pulso pós-sináptico de j , o peso dessa sinapse é incrementado porque este pulso exerce influência relevante na relação temporal. Sinapses cujos tempos de disparo são anteriores ou posteriores ao pulso de j são decrementadas em seus pesos, refletindo sua menor importância na geração do *spike* em j .

Com essa regra, padrões de entrada podem ser codificados nos pesos sinápticos tal que, após o aprendizado, o tempo de disparo de um neurônio de saída reflete a *distância* do padrão avaliado para o padrão de entrada aprendido [22].

Esta é uma regra de aprendizado não-supervisionado hebbiano. Essa regra é, em certo sentido, parecida com STDP¹⁴. Uma das hipóteses levantadas aqui é que a regra STDP pode definir a sinapse k que melhor ajusta o tempo entre i e j a fim de que um neurônio participe de uma assembleia.

A proposta é simples, porém têm um custo computacional e de memória a serem considerados: a partir do momento que um neuroide i dispara, seu *spike* se propagará paralelamente por todas as m sub-conexões até o neuroide j . Assumindo que cada sub-conexão começa com o mesmo peso sináptico, e que cada terminal sináptico $k+1$ tem, por exemplo, um retardo fixo de 1 ms em relação ao terminal k , então temos o *spike* chegando por m caminhos com 1 ms de atraso em cada sub-conexão. Assim, não só temos que memorizar m valores de conexões sinápticas, como o algoritmo deve levar em conta os $m.\delta t$ atrasos dos *spikes* em cada uma das conexões. Com $m=10$, por exemplo, é possível simular a gama presumida para atrasos nas árvores dendríticas mais os atrasos nas sinapses discutidos acima. O custo da implantação desse algoritmo é que o sistema deve ter na memória um vetor

¹⁴ A regra foi proposta em 1998, na época em que se começava a falar em STDP.

com 10 pesos para cada sub-conexão por neuroide, além de ter que considerar um vetor, digamos $[1, 2, \dots, 9, 10]^T$, como o vetor atraso para as m conexões sinápticas de cada par de neuroides. Ou seja, o algoritmo deve calcular o efeito do retardo de cada *spike* nas m sub-conexões sinápticas do neuroides x_i para um neuroide y_j . Repetindo-se isso para todos os neuroides interconectados, uma rede que utilize esse algoritmo pode ser inviável para cálculo em tempo real.

Contudo, com esse artifício, as conexões sinápticas cujo tempo tem relevância na formação do *spike* acabam prevalecendo sobre as outras se aplicarmos a regra STDP às redes. Com a aplicação de STDP, os pesos das sub-conexões nas quais os *spikes* chegam distantes (adiantados ou atrasados) do *spike* pós-sináptico em j vão perdendo importância. Por outro lado, as sub-conexões cuja relação temporal pré- e pós-sináptica são muito próximas se fortalecem.

Portanto, há um ajuste, ou um *aprendizado*, que favorece a conexão com o melhor tempo de propagação entre os neuroides x_i e y_j , ajustando simultaneamente os dois fatores relevantes na formação de assembleias: o *peso sináptico* e o *retardo na propagação* entre neuroides.

Como acontece a *cooptação* de neuroides numa assembleia usando essa regra de ajuste de tempo? A figura 3.6 ilustra o conceito. Na figura foi usada uma assembleia síncrona (*synfire*) por ser de mais fácil compreensão e visualização, mas o conceito serve para coalizões poli-sincronizadas.

O axônio de cada neuroide x_i está conectado à entrada dendrítica de todos os neuroides y_j . Os pulsos de x_i (pré-) são disparados sincronamente e chegam a y_j depois de 23 ms. Assim, todos os neuroides do grupo y disparam 23 ms após o disparo do grupo x , caracterizando uma assembleia.

Um neuroide z que também possui conexão total com os neuroides y começa a disparar quase simultâneo ao grupo x , mas os spikes de z não chegam exatamente no momento em que a assembleia y dispara. Vamos assumir que cheguem adiantados.

O atraso *axonal* de z para y_j é de 15 ms (constante) e, digamos que inicialmente a conexão mais forte entre z e cada y_j seja w_{zy_4} (4 ms de atraso). Por isso os pulsos

ajustam os retardos de propagação em sua árvore dendrítica, fazendo um “ajuste fino” para incluir novos neuroides em sua assembleia. Ou seja, o neuroide z é cooptado pela assembleia Y para fazer parte da coalizão. Note-se que a operação na árvore dendrítica de cada neuroide da coalizão afeta apenas a conexão que cada um deles faz como o novo candidato.

Isso faz sentido em termos biológicos, porque os neurônios de uma coalizão são aqueles que disparam PAs que caracterizam a coalizão, e só eles podem *identificar* quais pulsos chegaram adiantados ou atrasados em sua árvore dendrítica em relação aos PAs que eles geram.

Além disso, considerando que as distâncias axonais são (quase) constantes, são as árvores dendríticas que apresentam plasticidade estrutural e modificam as posições das conexões sinápticas, como vimos anteriormente; e, portanto, são elas que podem fazer alterações que causam algum tipo de reajuste na estrutura e na relação temporal entre os neurônios.

Assim se explica a hipótese do aprendizado temporal em que se baseia esta tese, e que permite que os próprios neurônios ou neuroides de uma coalizão façam cooptação de novas unidades para a assembleia na qual participam. Em outras palavras, *neurônios das assembleias se adequam temporalmente para cooptar novos candidatos e fortalecerem as assembleias nas quais participam.*

Trata-se de uma forma de aprendizado sem supervisão, hebbiana, e como tal, este aprendizado requer que a rede esteja sujeita a estímulos; ou seja, a rede deve passar por experiências para que ajustes ocorram. Em outras palavras, *neurônios nas assembleias só conseguem cooptar outros neurônios se a rede passar por experiência. Mais ainda, se considerarmos todas as formas de plasticidade que ocorrem na rede, as assembleias se manterão apenas se continuarem sob estímulo e/ou constante re-estimulação das assembleias já formadas.*

Por enquanto, intuímos que esta regra de aprendizado não-supervisionado seja suficiente para auto-organização na computação por assembleias. Mas é prematura qualquer conclusão a esse respeito. Paugam-Moisy e Bohte [22] descrevem e fazem revisão de trabalhos de aprendizado supervisionado aplicado a RNPs; por exemplo, aplicando regras de *error-backpropagation*. Pode ser interessante, talvez necessário,

o uso de um método supervisionado de aprendizado temporal na abordagem das assembleias. Contudo, pelo exposto acima, parece que o método hebbiano não-supervisionado nos deixa próximo de obter sucesso com simulações de coalizões, além de conseguir explicar como neuroides são cooptados pelas assembleias, e como a conexão, cujo retardo temporal se torna a mais fortalecida, leva a conexões concorrentes e, possivelmente, ao desaparecimento.

A hipótese acima exposta nos permite inferir que uma assembleia, que pode começar com um núcleo pequeno, funciona como um *atrator* na rede no sentido de agregar todos os neurônios/neuroides relevantes em sua vizinhança que estejam pulsando dentro de uma faixa temporal condizente.

Some-se ao exposto nos tópicos anteriores (plasticidade estrutural e computação dendrítica) e temos uma justificativa biologicamente plausível para as múltiplas conexões assumidas acima, assim como para os ajustes temporais que acontecem entre os neuroides, e isso ocorrendo sem necessidade de um aprendizado supervisionado.

Dessa forma, pelo menos neste estágio da pesquisa, podemos descartar a necessidade de inserir regras de aprendizado supervisionado à computação por assembleias, ficando este tópico como possibilidade para investigações futuras, talvez para funções cognitivas ou de inteligência em mais alto nível.

3.8 Resumo do capítulo

Duas características primordiais das máquinas cerebrais são robustez e flexibilidade. Sistemas nervosos modificam suas respostas comportamentais, eles aprendem a partir da experiência. SNs modificam suas propriedades internas de sorte que continuam operacionais em condições diversas.

Usa-se o termo *plasticidade*, de um modo geral, em referência à propriedade do cérebro de se modificar e se adaptar às demandas computacionais a que estão sujeitos. Na verdade, toda plasticidade que ocorre no cérebro ocorre de fato nas unidades computacionais, nos neurônios.

Neurônios modificam propriedades intrínsecas para se manterem dentro de parâmetros operacionais apropriados para as tarefas que executam. Neurônios também modificam o fator de acoplamento entre si, modificando assim as topologias das redes que formam. No nível da célula neural, podemos considerar dois tipos de plasticidade. Uma é a plasticidade de curto-prazo: neurônios submetidos a rajadas de disparos precisam diminuir o efeito da sucessão de disparos para que não entrem em colapso, ou seja, os PEPS e PIPS subsequentes numa rajada devem diminuir sua influência sobre a célula para não saturá-la. Este tipo de plasticidade ocorre num intervalo de tempo bem curto.

A outra forma de plasticidade intrínseca da célula é a plasticidade homeostática: neurônios com atividade excessiva, ou sem atividade alguma, ativam mecanismos que controlam sua sensibilidade ou sua excitabilidade de forma que permaneçam sempre numa região apropriada para atividade, ou seja, eles se mantêm funcionalmente prontos para operar.

No nível das relações entre neurônios temos que considerar um tipo de plasticidade diferente e importante. Neurônios modificam o fator de acoplamento sináptico por meio de vários mecanismos, fortalecendo as conexões sinápticas por períodos longos (LTP) ou enfraquecendo as sinapses sem uso (LTD). Há evidências de que este mecanismo é responsável pelo aprendizado e formação de memória. Conhecido como aprendizado hebbiano, estes mecanismos estão atualmente sendo desvendados em termos da relação temporal entre os *spikes* que atingem a fenda pré-sináptica e os *spikes* gerados pela unidade pós-sináptica.

Esta regra de plasticidade tem sido genericamente denominada STDP (*spike-timing dependent plasticity*). Sua utilização em redes pulsadas tem se mostrado eficaz para implantação da “regra hebbiana temporal”, que ajusta os pesos sinápticos em longo prazo, em função da experiência da rede, levando em conta a relação temporal entre disparos pré- e pós-sinápticos.

Assim como no caso da modelagem dos neurônios, nas simulações em RNPs é preferível modelar o *fenômeno* da plasticidade em vez de descrever no modelo detalhes fisiológicos que ocorrem dentro das células ou nas sinapses. O capítulo mostra as abordagens e algoritmos usados para implantação de cada uma das

plasticidades discutidas, bem como suas implicações na abordagem da computação por assembleias.

É preciso aprofundar estudos que mostrem como as regras de plasticidade interagem entre si, e as implicações dessas interações na função computacional da rede ou do SN como um todo. Assim, há um tópico mostrando alguns trabalhos que tentam integrar todas as regras de plasticidade em simulações de RNPs.

Computação por assembleias, da forma proposta nesta tese, está baseada em dois fatores que não podem ser independentes: o *fator de acoplamento* entre unidades computacionais e os *atrasos na propagação* entre unidades. Ou seja, no *peso sináptico*, que define a topologia, e no *atraso na propagação*, que é o fator que promove as coincidências temporais que permitem o surgimento de coalizões. A questão do peso sináptico pode ser resolvida pelas plasticidades celulares, a STDP associada às outras duas plasticidades intrínsecas, que modificam topologias e que mantêm as unidades operacionais, respectivamente.

Por outro lado, deve haver algum tipo de ajuste nos tempos de propagação de forma que aumentem as possibilidades de formação de assembleias. Ou seja, se acontecerem pequenos ajustes temporais nos atrasos entre unidades, esses neurônios/neuroides poderão ingressar ou serem excluídos de outras assembleias. Assim, a pertinência ou não de uma unidade nas assembleias passa a ser um fator sinérgico que depende do ajuste de atrasos junto com os pesos sinápticos. Ou seja, pequenas mudanças nos atrasos entre unidades passam a ser componente computacional que pode representar uma memorização, uma representação, etc., como será discutido adiante no capítulo sobre formação de assembleias. Mas, como podem ocorrer tais “adaptações temporais”?

Durante o capítulo foi mostrado que uma das formas de plasticidade que tem sido discutida atualmente é a plasticidade estrutural ou anatômica. Descobriu-se que o córtex não tem uma estrutura de conexões fixa (*hardwired*), mas que sinapses são renovadas e desaparecem em função de estímulos. A maleabilidade no córtex envolve tanto a plasticidade celular (intrínsecas e hebbianas) quanto a plasticidade estrutural, que ocorre a partir da modificação das posições das conexões sinápticas.

A árvore dendrítica não é um lugar passivo, arborizado apenas para permitir que novas conexões surjam e desapareçam. Há tempos questiona-se o poder computacional da árvore dendrítica, em cuja ramificação uma série de funções lógicas e operações acontecem. Por isso, foi incluído um tópico sobre possibilidades de computação na árvore dendrítica e como elas podem ser implantadas em RNPs.

Há evidências advindas de experimentações de que em árvores dendríticas ocorrem pelo menos: operações lógicas (computação de funções *AND*, *OR*, *NOT*), detecção de coincidências, classificação por atenuação e amplificação, além de os dendritos funcionarem como filtro atenuador passa-baixas.

Uma das hipóteses levantadas nesta tese é que algumas das plasticidades que ocorrem na árvore dendrítica contribuem para modificação da relação temporal entre unidades. Assim, além das computações executadas na árvore dendrítica, ocorre também reajuste da temporização entre unidades.

A troca de posição física das sinapses na árvore dendrítica pode ser o mecanismo capaz de explicar tal ajuste temporal, uma vez que modificando a posição física das sinapses, modifica-se o tempo de propagação dos *spikes* na árvore dendrítica. Discutiu-se ainda a relevância dessas capacidades funcionais na computação por assembleias.

A contribuição maior do efeito computacional dendrítico em assembleias parece estar justamente no ajuste de atrasos temporais, ou seja, na facilitação de uma unidade computacional participar de mais assembleias por meio de pequenos ajustes nos tempos de propagação dos *spikes* de seus vizinhos. Dois algoritmos que executam este “aprendizado temporal” foram discutidos ao final do capítulo.

Este capítulo serviu de base para justificar a necessidade de implantação das regras de plasticidade celulares nas simulações, bem como justificar duas hipóteses principais desta tese: *“neurônios das assembleias se adéquam temporalmente para cooptar novos candidatos e fortalecerem as assembleias nas quais participam num processo de auto-organização”*.

Some-se a esta, a hipótese de que: *“as assembleias só conseguem cooptar outros neurônios se a rede passar por experiência. Considerando todas as plasticidades*

que ocorrem nos neurônios, as assembleias se manterão apenas se continuarem sob estímulo e/ou constante re-estimulação das assembleias existentes.

O exposto acima ilustra nossa visão de que assembleias e a computação que delas advém são fenômenos corporificados nas redes sob constante estímulo, interagindo com o meio e dependendo dessa interação e constante estímulo para manterem-se operacionais.

4 REDES E ASSEMBLEIAS

//

No capítulo *II* foi apontado que existem níveis de abordagem quando pensamos em computação cerebral. No capítulo anterior foi tratado o nível das células e das modificações intrínsecas que nelas acontecem. Neste capítulo vamos tratar do nível acima: o nível da inter-relação entre as unidades. As perguntas que guiam este capítulo podem ser: o que acontece quando neurônios são conectados uns aos outros? Como a interação entre eles gera operações computacionais?

Primeiramente se faz uma investigação sobre o que um neurônio sozinho pode fazer em termos computacionais. Depois, são analisados os tipos de conexões entre neurônios e a relação com retardo na propagação e fator de acoplamento entre as células; dois dos fatores importantes que interferem na comunicação intercelular, consequentemente, na computação que executam.

RELEVÂNCIA PARA A TESE:

Analisa-se aqui como um neurônio ou grupos deles podem gerar computação, quais fatores influem na interconexão de células neurais e como isso afeta a computação que os circuitos executam. Discutimos pequenos circuitos e as possibilidades das redes com grande número de neuroides. Mostramos como assembleias neurais se formam e como se tornam biestáveis, com capacidade para representar ou controlar algoritmos. Dessa forma, o capítulo provê a base para compreensão da computação por assembleias biestáveis.

Reunir células neurais significa compor ou configurar circuitos neurais, e a forma como circuitos se conectam (as topologias) é determinante para as operações computacionais realizadas por redes de neurônios. Assim, discute-se a formação de topologias como CPGs (*central pattern generator*), redes com capacidade de aprendizado e potencial computacional “aberto”, e redes determinísticas, caso das assembleias biestáveis neste estágio da pesquisa, como proposto adiante.

Ainda que se formem circuitos determinísticos (ou seja, com conexões pré-estabelecidas), neurônios biológicos estão sujeitos a alterações (plasticidades) intrínsecas impelidas pelo regime de operação a eles impostos, assunto do capítulo anterior. Da mesma forma, as sinapses modificam-se em função da quantidade de ativação que sofrem no tempo. Estes dois aspectos modificam a computação da

rede como um todo. Em outras palavras, não só os neurônios devem ser vistos como sistemas dinâmicos, mas cada circuito que se forma com neurônios também se comporta como sistemas cuja dinâmica se modifica dependendo da forma de ativação. *Uma RNP é um sistema dinâmico criado a partir da interconexão de sistemas dinâmicos*, ou seja, são sistemas com acentuado grau de complexidade. Dessa forma, vamos discutir os tipos de respostas que emergem de sistemas complexos dessa natureza, o que se pode fazer para restringir os limites operacionais e, por conseguinte, obter assembleias neurais operando com algum grau de previsibilidade; bem como mostrar possíveis caminhos para se incluir algum grau de modulação às redes, papel desempenhado por neurotransmissores.

4.1 Redes neurais pulsadas recorrentes

Independente de serem biológicas ou artificiais, com uma ou mais camadas, é possível destacar duas classes principais de topologias de redes neurais: as redes de *propagação unidirecional*¹⁵ (*feed-forward*) e as recorrentes (com realimentação).

Nas redes de propagação unidirecional, ou sempre adiante, o sinal percorre a rede em uma única direção, passando por uma ou mais camadas de neurônios (ou neuroides). Todas as conexões apontam num único sentido: das camadas anteriores para as posteriores. Nesse tipo de rede, os neurônios/neuroides da mesma camada não são conectados entre si (lateralmente), nem os das camadas posteriores se conectam com as camadas anteriores (retorno); dessa forma, o sinal tem apenas um sentido de propagação: da entrada para a saída.

Por outro lado, as redes neurais recorrentes (RNR) possuem as saídas de alguns neurônios/neuroides conectadas a unidades da mesma camada (inclusive a eles próprios), ou conectados a unidades de camadas anteriores. Estas conexões criam um ou mais laços de retroalimentação, o que faz com que os sinais (PAs ou *spikes*) percorram a rede em ambas as direções: da entrada para a saída e vice-versa,

¹⁵ Talvez a melhor tradução para *feed-forward propagation* seja propagação unidirecional. Contudo, Timoszczuk [71] utiliza o termo *propagação para adiante* em sua tese.

criando memória dinâmica na rede. Devido ao fato de possuírem laços de *feedback*, RNRs têm a capacidade de processar informação recursiva temporal.

Por operarem com transmissões cíclicas de sinais, as RNRs podem desenvolver dinâmicas de ativação temporal auto-sustentadas, mesmo quando não são ativadas por sinais de entrada. Além do mais, quando acionadas por estímulos, RNRs podem manter em estados internos uma transformação não linear da história de entrada. Portanto, matematicamente, RNRs se comportam como sistemas dinâmicos, enquanto as redes neurais *feed-forward* se comportam como funções [8], [107].

O treinamento de RNRs é difícil, em especial as criadas com neurônios pulsantes (RNPR redes neurais pulsantes recorrentes). Visando solucionar este problema, recentemente surgiram propostas de construções de RNR randômicas (pulsadas ou não), associadas a uma camada de neuroides de saída. A inovação é que treina-se apenas a camada de neuroides lineares na saída, chamada *Readout Layer*. A terminologia “*reservoir computing*” tem sido usada como um nome genérico para este tipo de computação em máquinas neurais.

A metodologia da *reservoir computing* (traduzindo livremente, computação num reservatório) pode ser implementada tanto em redes pulsadas (chamadas *Liquid State Machines* – LSMs – apresentadas em [108]) quanto em redes não pulsadas (chamadas *Echo State Machine* – ESMs – apresentadas em [109]). O termo “*reservoir*” refere-se ao conglomerado (*pool*) de neuroides interconectados por um conjunto de ligações e pesos sinápticos randômicos e esparsos, que formam uma espécie de reservatório (revisão em [9], [110], e subsequentes referências).

Numa visão ampla sobre esta abordagem, pensa-se que o córtex é como um grande reservatório onde sinais reverberam constantemente e apenas as camadas de neurônio de entrada e saída são treinadas para extrair ou inserir dados no reservatório. A abordagem *reservoir computing* está bastante ativa atualmente.

A estrutura proposta nesta tese é uma RNR organizada em pequenos agrupamentos (*pools*) modulares. Assembleias se formam porque neurônios possuem laços recorrentes, como veremos adiante. Os *pools*, por sua vez, são formados por núcleos (*kernels*) ainda menores; portanto, *pools* são também redes recorrentes de neuroides. Apesar de ambos, *pools* e *kernels*, serem RNR, eles podem ser

associados tanto às redes *feed-forward* quanto a outras redes recorrentes, ou ainda a máquinas *reservoir*. A condição necessária é que estas redes devem ser implementadas por redes de neuroides pulsantes que levam em conta o tempo de atraso na propagação.

Redes neurais recorrentes representam uma ampla e variada classe de modelos computacionais usados principalmente por modelagem e simulação de “cérebros biológicos” ou como ferramenta de engenharia para aplicações técnicas. De acordo com Lukoševičius e Jaeger [107], a partir de uma perspectiva de sistemas dinâmicos, há duas classes principais de RNRs: a primeira classe tem modelos que são caracterizados por dinâmicas estocásticas de minimização de energia e de conexões simétricas (ex: redes de Hopfield). A segunda classe tipicamente retrata dinâmicas determinísticas e conexões direcionadas.

As redes usadas neste trabalho (*kernels* e *pools*) se encaixam na segunda classe, uma vez que são projetadas ou desenhadas *a priori*, e até este estágio da pesquisa, não foram estudados algoritmos de aprendizado e auto-organização para assembleias. Considera-se nesta tese que a distância entre neuroides (distância axônica) determina os atrasos mais marcantes na propagação entre neuroides, ficando o ajuste fino (adaptação temporal) para a plasticidade na árvore dendrítica, como pode ser visto no capítulo anterior.

4.1.1 Estratégia para simulação de RNPs

Há duas famílias de algoritmos mais usados para simulação de redes neurais pulsadas: modelos *síncronos* e modelos ativados por *eventos*. De acordo com [23], podemos classificar as redes em *síncronas*, impelidas pelo relógio do sistema (*clock-driven*), nas quais todos os neuroides são atualizados simultaneamente a cada iteração computacional da simulação. As outras redes são as *assíncronas*, impelidas por eventos (*event-driven*), nas quais os neurônios são atualizados apenas quando recebem ou emitem um *spike*.

Existem redes de estágios híbridos que englobam ambos os métodos, mas não será o caso aqui. Neste trabalho, optou-se por redes ativadas por eventos, pela própria

natureza das formações e desativações de assembleias de acordo com os gatilhos que as ativam. Assim, as redes neurais nesta tese cuidam de uma computação no tempo, ativadas por eventos, nas quais a “codificação” de dados e o “controle” advêm da temporização entre *spikes* e da formação de assembleias.

Essas duas abordagens têm algumas características em comum, pois ambas podem ser modeladas em tempo contínuo ou discreto. No caso das assembleias, optou-se por modelagem, simulação e computação em tempo discreto, uma vez que é da natureza da computação proposta que os eventos sejam fenômenos discretos, por exemplo, o acontecimento de uma coalizão, a detecção de um padrão etc. Baseado em experiências e simulações relatadas na literatura [90], [91]), a melhor resolução temporal para o sistema é de 1 ms, que equivale ao menor evento a ocorrer na rede: um *spike*.

Levando-se em conta estas características gerais usadas para as simulações, vamos analisar que tipos de redes levam à formação de assembleias, começando com circuitos simples, com um ou dois neurônios acoplados, para então analisar a formação de circuitos mais complexos, com vários neurônios, até grupos sincronizados, assíncronos e multiplamente sincronizados.

4.2 Marca-passo (*pacemakers*) e osciladores

Neurônios sozinhos podem gerar atividades rítmicas, emergentes de propriedades intrínsecas associadas a influências externas. Características internas (como influxo de íons dependentes de tensão/tempo, segundo mensageiros, íons de Ca^{++} intracelular, etc.), em associação com moduladores (neurotransmissores), além de entradas de sinais pelas conexões sinápticas, determinam o comportamento oscilatório de um neurônio isolado.

Como resultado, eles disparam repetidamente potenciais de ação numa frequência fixa (ver Fig 3.1A: *class 2 excitable*). Podem também disparar rajadas intermitentes com períodos silentes (*tonic bursting*), ou ainda podem produzir ondas abaixo do limiar de disparo (*subthreshold oscillations*). Isso pode ocorrer mesmo na ausência de entradas sinápticas independentemente da anatomia ou topologia da rede.

Estes neurônios são conhecidos como *osciladores endógenos* (*endogenous oscillators*) ou como neurônios marca-passo (*pacemakers*) [3], [111], [112]. Do ponto de vista de sistemas dinâmicos, os estados desses neurônios giram em torno de um ciclo-limite, ou uma órbita periódica estável. No modelo neural simples de Izhikevich, a alteração no valor de alguns parâmetros pode transformar o neuroide num oscilador marca-passo, ou qualquer dos acima citados.

Figura 4.1 – **Marca-passos e osciladores**. (A) Neurônios isolados podem apresentar comportamento oscilatório (*pacemakers*), como pode ser visto nas formas de ondas acima. (B) Circuitos neurais também podem ser responsáveis por gerarem oscilação interna.

Outra propriedade dos neurônios *pacemakers* é que eles podem alternar de um estado para outro, geralmente denominado como multiestabilidade ou múltiplos estados estáveis (*multistability* [113]). A figura 4.1A mostra uma célula *pacemaker* e algumas formas de onda geradas por neurônios isolados funcionando nestes estados operacionais.

Neurônios pacemakers foram descritos acima porque eles podem executar um papel importante para computação por assembleias biestáveis: *eles podem favorecer a sincronização ou facilitar a formação das assembleias*. Na verdade, eles são elementos importantes nos circuitos de redes rítmicas em geral.

Atividade rítmica padronizada pode ser gerada não apenas por neurônios marca-passo, mas também por circuitos com conexões sinápticas específicas (*hardwired*) entre neurônios que não são osciladores espontâneos.

Por exemplo, neurônios podem responder a um simples disparo isolado quando excitado e ainda assim ser parte de um circuito oscilador, como o circuito apresentado na figura 4.1B. Neste circuito, dois neurônios se excitam mutuamente, formando uma rede osciladora (*network oscillator* [112]). Um pulso do neurônio *a* provoca o disparo (*triggering*) no neurônio *b*, que retroalimenta o neurônio *a*. Nesta tese, quando houver referência a *oscilação interna*, está-se referindo a sinais que podem ser gerados tanto por meio de *pacemakers* quanto por meio de grupos de neurônios (ou até mesmo populações) que se excitam ou se inibem mutuamente, gerando sinais cíclicos.

4.2.1 Redes de osciladores e CPGs

O próximo passo no sentido dos circuitos neurais mais complexos pode ser conectar osciladores endógenos com sinais (pulsos) gerados externamente. Quando um ou mais neurônios são conectados com uma fonte de pulsos, outros fatores podem afetar a dinâmica da rede, especialmente o atraso de propagação do pulso pela rede. A figura 4.1B mostra uma montagem típica de uma assembleia mínima, a *assembleia dual*: que pode ser um circuito recorrente excitatório ou inibitório. Em neurônios estes circuitos podem apresentar uma complexidade extraordinária, além de comportamento *multiestável*, mesmo em “simples” invertebrados [113].

Redes de osciladores¹⁶ e *pacemakers* quando acoplados a fontes de pulsos externos formam osciladores acoplados por pulso (*PCO – pulse-coupled oscillators*). Pulsos de entrada podem mudar o regime operacional nesses osciladores, por exemplo, podem alterar a fase do próximo pulso; assim, a rede ou um neurônio isolado pode apresentar comportamento multiestável [74]. Geralmente, o método mais usado para analisar e rastrear quanto um pulso de entrada avança ou atrasa o próximo pulso é a curva de resposta de fase (*PRC – phase response curve*) (análise de diferentes redes PCO em [114]).

Estudos no nível neural mostram que propriedades intrínsecas da membrana, modulação neural, e ativação sináptica são responsáveis por gerar diferentes padrões de atividade nestes circuitos. A natureza construiu vários PCOs como soluções biológicas para geração de padrão motor em animais “simples”. Redes de sistemas osciladores estão amplamente em atividade tanto em invertebrados quanto em vertebrados (ver [115], [113]).

Como foi dito, RNP dedicadas à geração de padrões rítmicos são chamadas CPGs (*central pattern generators*, revisões em [115], [116], [117]). Há inúmeras combinações anatômicas dos neurônios que executam geração de atividades padronizadas. Em organismos, por exemplo, o nado no molusco *Clione limacina* e o bater do coração em sanguessugas são gerados por um circuito de *inibição recíproca por interneurônios* [115], [112]; enquanto o ritmo pilórico dos gânglios estomogástricos em crustáceos é um padrão trifásico [118], similar ao do sistema respiratório dos mamíferos [115]. Na verdade, o gerador de padrões respiratórios nos mamíferos é formado por uma combinação de CPGs [119]. Deveríamos considerar um gerador de atividade padrão (ou circuitos oscilatórios) como assembleias neurais?

Tecnicamente eles podem ser considerados assembleias, uma vez que os neurônios operam em conjunto, codificando ou controlando ações usando o princípio de “correlação e sincronismo”, uma das formas de codificação temporal nos *spikes* (ver

¹⁶ Certamente redes osciladoras de dois ou três neurônios geram padrões de disparos, e tal atividade é chamada *pattern generation*. Contudo, preferiu-se aqui usar CPG apenas para circuitos mais complexos, que geram padrões encadeados, como na atividade motora, cardíaca, respiratória, etc.

tópico 2.5 item c). Mas estes circuitos são geralmente *hardwired*, são dedicados, e isso os diferencia da formação efêmera que ocorre nas assembleias.

Assembleias, como é proposto adiante, ocasionalmente formam atividades que se parecem com padrões de CPGs, mas são fenômenos que podem se desmanchar tão rapidamente quanto se formam, em decorrência da função da operação computacional que está sendo executada. Por outro lado, os CPGs são circuitos perenes, que permanecem gerando padrões de atividades, mesmo que sob comando de circuitos superiores, como os CPGs motores. É possível que a dinâmica cerebral faça uso de muitos padrões gerados por CPGs para formar coalizões.

Portanto, faz-se nesta tese uma diferenciação qualitativa entre CPGs e as assembleias, no sentido de que os CPGs são circuitos mais perenes enquanto coalizões são dinâmicas. CPGs são entendidos como circuitos fixos que geram padrões de atividade neural estereotípica, embora esta visão esteja se modificando; por exemplo, Yuste e colaboradores [120] sugerem que circuitos no *neocórtex* evoluíram a partir de circuitos rítmicos motores e de CPGs, originalmente presentes na medula espinhal e tronco cerebral. Essa é uma visão revolucionária que faz muito sentido quando pensamos no processamento por assembleias.

Os autores apontam similaridades entre os circuitos corticais e os circuitos de CPGs na anatomia, na biofísica, nas patologias, na modularidade, na dinâmica; embora a principal diferença esteja na plasticidade dos circuitos: circuitos do córtex são muito mais plásticos. Em outras palavras, “[o córtex pode ser formado por] um tipo especial de CPG baseado em assembleias hebbianas específico para aprendizado e armazenagem ou recuperação de memórias (um ‘CPG aprendiz’ ou ‘CPG memória’)” [120] (tradução nossa, original no rodapé)¹⁷. Repetindo, essa é uma hipótese importante que vem ao encontro de nossa abordagem, uma vez que descrevemos assembleias neurais biestáveis como dependentes de alguma organização estrutural, sendo este um papel que pode ser executado por algum tipo de CPG multifuncional.

¹⁷ “a special type of CPG that is based on Hebbian assemblies and is specialized for learning and storing or retrieving memories (a ‘learning CPG’ or ‘memory CPG’)” [120].

Em outras palavras, em nossa visão assembleias biestáveis podem até formar CPGs temporários, uma vez que elas reverberam entre si, e dessa interação o cérebro pode obter representação e computação. Portanto, estamos próximos da hipótese de que o neocórtex evoluiu dos CPGs e que no neocórtex ocorrem CPGs com maior plasticidade. Assembleias biestáveis são, em certo grau, CPGs efêmeros, que nascem e desaparecem de acordo com a necessidade e a operação em execução; isso é regido por algum tipo de estrutura, que bem pode ser alguns tipos de CPGs fixos, que marcam passo ou ditam o controle rítmico.

Antes de discutimos formação de assembleias em si, apenas mais um assunto deve ser ponderado, ou seja, a questão das frequências das oscilações em CPGs e nas coalizões, uma função quase que exclusiva dos retardos na propagação dos pulsos entre neurônios ou neuroides.

4.3 Atrasos e frequências

De acordo com Marder [115], um aspecto importante na geração de padrão motor é saber o que controla a frequência do sistema. Também importante é estabelecer se o mesmo grupo neural pode ser *reconfigurado* para gerar diferentes padrões motores, ou se cada resposta comportamental requer uma rede dedicada. Além disso, devido à quantidade e aos diferentes tipos de CPGs (e *pacemakers*), a contribuição de cada circuito gerador de padrão para o resultado final da operação da rede é impossível de ser predeterminado [111]. Mais ainda, crucial para a dinâmica da rede é o atraso na propagação dos pulsos entre as unidades.

Atrasos nas redes neuronais biológicas advêm, como já discutimos no capítulo anterior, da propagação ao longo dos axônios nas fendas sinápticas e nas árvores dendríticas. A velocidade de condução para axônios *mielinados* em mamíferos varia e é determinada experimentalmente. Por exemplo, Swadlow [69] registrou a velocidade de 1 m/s em fibras cortico-corticais de coelhos, Bear apresenta 10 m/s [6] p.94, e Kandel fala em taxas de 1-100 m/s [3] p.21. Em Purves pode-se ler: “for example, whereas unmyelinated axon conduction velocities range from about 0.5 to 10 m/s, myelinated axons can conduct at velocities of up to 150 m/s” [5] p.63. Em [121], há duas curvas demonstrando a relação entre diâmetro dos axônios

(mielinados ou não) versus velocidade de propagação; nelas se lêem velocidades axonais de até 120 m/s. Em contraste, a velocidade de condução em axônios não-mielinados pode ser tão baixa quanto 0,15 m/s, e isso pode significar atrasos de até 10 ms em circuitos relativamente curtos do córtex [90].

O que se quer é determinar uma faixa de atrasos biologicamente plausível para aplicarmos nas simulações de assembleias. Se escolhermos atrasos não muito reais e muito grandes, por exemplo, podemos obter um incremento na capacidade de formação de assembleias, mas o custo computacional aumenta porque uma janela de tempo de atrasos tem que ser vasculhada. Por outro lado, atrasos muito curtos aumentam a velocidade da computação, mas podem restringir a capacidade total de formação de coalizões. Para encontrar um balanço, melhor é recorrer aos circuitos que sabidamente funcionam, e usar uma abordagem bioinspirada.

Atrasos de propagação foram progressivamente introduzidos nas RNPs [22] e têm se mostrado cruciais na determinação do comportamento dinâmico das redes [74], [122]. Várias simulações de redes neurais pulsadas não consideram atrasos entre um pulso que acaba de ser gerado em um neuroide e sua ação no neuroide ou na camada seguinte; ou seja, na iteração seguinte, o pulso do neurônio pré- já provoca efeito nos neuroides pós-sinápticos aos quais ele está conectado.

Contudo, neurônios estão fisicamente distantes uns dos outros e o atraso de propagação existe, sendo um elemento efetivo na computação executada; assim, o tempo em que um pulso atua em outros neurônios aos quais ele está conectado é fundamental; é uma característica determinística.

Dessa forma, no nosso entendimento, nas RNPs os atrasos de propagação devem ter tanta importância quanto o valor dos pesos sinápticos, das características da membrana que promovem plasticidade, ou dos neurotransmissores e seu papel modulador. Como foi dito e merece ser enfatizado, *retardos* e *pesos sinápticos* são os componentes mestres na formação de coalizões. Os últimos moldam a topologia, e os primeiros limitam a frequência máxima de operação.

Certamente vamos observar oscilações em assembleias biestáveis; embora como *consequência* da reverberação entre coalizões, não como a *causa* para que as assembleias se formem. Eis porque CPGs e pacemakers são importantes para

gerarem ritmos que favorecem a ocorrência e sincronização de assembleias. Mas tanto em CPGs quanto em assembleias o atraso na propagação é determinante para a frequência da oscilação ou reverberação entre grupos de neurônios ou neuroides.

Vamos analisar a influência do atraso num circuito oscilador simples, como o mostrado na figura 4.1B. Seja δf o tempo de disparo do *spike* e δr o período refratário da tensão da membrana, que podem ser considerados constantes para um dado tipo de neurônio. Seja δpij o atraso na propagação do *spike* de um neuroide i para um neuroide j ; assim, as duas células formam um loop que pode reverberar numa frequência calculada por:

$$F = \frac{1}{(\delta_{fA} + \delta_{rA} + \delta_{pAB} + \delta_{fB} + \delta_{rB} + \delta_{pBA})}$$

onde δ_{fA} denota o tempo de duração do *spike* do neuroide a (fig.4.1), δ_{rA} o período refratário de a e δ_{pAB} o tempo de propagação do *spike* de a para b ; e ainda, δ_{fB} denota a duração do *spike* b , δ_{rB} o período refratário de b e δ_{pBA} o atraso na propagação de b para a .

Considerando a duração do *spike* um tempo fixo de 1 ms¹⁸; considerando ainda que os neurônios são idênticos, então eles apresentam o mesmo período refratário. Mais ainda, se os comprimentos dos axônios que os conectam forem iguais, diremos que estão em distâncias simétricas, então eles possuem atrasos idênticos, assim a frequência poderá ser calculada por:

$$F = \frac{1}{2(\delta_f + \delta_r + \delta_p)}$$

Agora, vamos supor que enquanto o neurônio a estiver no estágio de recuperação, seu *spike* estará se propagando até b , portanto, $\delta p = \delta r$. Neste caso, a propagação

¹⁸ Neurônios geram PAs aproximadamente idênticos em forma e duração. Este tipo de afirmação era uma máxima, e.g. sobre PAs: “Their shape is relatively constant across species and types of neurons [...] the entire sequence occurs within less than 1ms” (Axonal Modeling, Koch&Bernander, em [4] p.135). Porém isso tem sido contestado: “The shape of action potentials differs considerably among various types of neurons in the mammalian brain” [139]. Cauli et al.[140] mostram que diferentes células corticais têm *spikes* que duram entre 1,5 e 2,6 ms. Alle et al.[141], analisando a eficiência energética de PAs em ratos, mostram que *spikes* podem durar pouco mais que 250µs. Para fins de simulação, o tempo de 1 ou 2 ms tem sido utilizado, e.g. [sobre *spikes*]: “This brief electric pulse (1 or 2 ms duration) then travels along the neuron’s axon, that in turn is linked up to the receiving end of other neurons, the dendrites” [22].

mínima de 1ms pode ser considerada entre neurônios. Seguindo [46], 1 ms será usado para a resolução temporal nas simulações, um tempo biologicamente plausível, embora seja possível mudar esta resolução. Assim, o *loop* formado pelos neurônios opera à frequência de $F=1/2(\delta_f+\delta_p) \equiv 1/4 \text{ ms} \equiv 250 \text{ Hz}$. Neurônios pulsando a esta frequência, ou próximo, são frequentemente citados na literatura descrevendo sistemas biológicos reverberantes [76], especialmente durante crises de epilepsia [123].

A interpretação pode ser: quando neurônios estão suficientemente perto e ocorrer o mínimo de atraso de propagação (dentro do período de refração do neurônio emissor do *spike*), estes loops recorrentes operam realmente rápido. Na medida em que se afastam uns dos outros, ou seja, axônios e árvores dendríticas se alongam e introduzem retardos, os atrasos inter-neurônios δ_{pBA} e δ_{pAB} ganham importância e determinam a frequência máxima. Considerando-se que não há informação na duração do *spike*, como sugerem [28], e que eles duram um tempo fixo de 1 ou 2 milissegundos, é possível criar simulações desconsiderando estas constantes, ou seja, voltar o foco apenas para os atrasos axonais e os dendríticos, ou resumi-los em um único atraso inter-neurônios.

Em suma, nos modelos e simulações de assembleias devemos considerar a duração dos *spikes* como um tempo fixo, geralmente 1 ou 2 ms. Cada *spike* gasta um tempo para atingir a sinapse de outro neurônio, e esse retardo é considerado na computação por assembleias. Coalizões acontecem justamente porque existem diferentes retardos na propagação. Na visão assumida nesta tese, atrasos têm uma relação direta com a distância entre neurônios (axônios + dendritos), ou relação com a soma dos atrasos do caminho pelo qual um *spike* percorre (em conexões com múltiplos neuroides).

Podem acontecer casos em que os neurônios/neuroides tenham um tempo refratário longo e estejam muito próximos uns dos outros. Se o atraso for mínimo e os pulsos chegarem aos neurônios seguintes ainda dentro do período refratário absoluto destes, não haverá reverberação alguma. Caso cheguem durante o período refratário relativo, a reverberação acontecerá se a excitação for suficientemente forte, ou seja, se houver muitos neurônios simultaneamente contribuindo para a geração do PEPS, ou se os acoplamentos sinápticos forem muito fortes, como é o

caso das sinapses elétricas (*gap-junctions*). Fora esta situação, o atraso de propagação entre neurônios será mandatório e *spikes* em si representam um tempo de duração bem menor que a escala de tempo na qual as assembleias reverberam.

Santos e colaboradores [124], realizando trabalho reverso de identificação de formações de coalizões, destacam que janelas de 30 ms são potencialmente bem ajustadas para análise de assembleias e que outros trabalhos já demonstraram que algumas assembleias celulares tendem a uma sincronização transiente em janelas de tempo de 25 ms. Ou seja, a soma de atrasos axonal mais dendrítico resultará biologicamente plausível se assumirmos uma faixa de 1-30 ms.

4.4 Formação de assembleias

Grupos de neurônios formam assembleias e operam em conjuntos. Às vezes usamos como sinônimo de assembleia o termo coalizão. Esta ideia não é nova. Na década de 1930, Sherrington já discutia a formação de *pools* de neurônios motores, enquanto Hebb, em seu livro *The Organization of Behavior* (1949) introduziu o conceito de assembleia de células e postulou os mecanismos de percepção e aprendizado (ver revisão em [125]). Mountcastle, por sua vez, introduziu o conceito de “coluna cortical” em 1957 [126], descrevendo neurônios das camadas II a VI do córtex visual de gatos, que respondem a estímulos simultâneos advindos de células receptoras do mesmo campo visual, formando grupos que atuam conjuntamente.

RELEVÂNCIA PARA A TESE:

Apresenta-se a seguir o conceito de assembleias e como elas se formam; o que são cadeias *synfire chains*, *synfire braid*, grupos policronizados. Além disso, na sequência discute-se a formação de assembleias biestáveis ou grupos policronizados biestáveis, base para formação de memória e algoritmos.

Assembleias podem ainda ser inferidas da análise de leituras experimentais ou por análise matemática de uma rede de componentes com conexões sistemáticas. Por exemplo, é possível medir a correlação cruzada entre vários trens de pulsos e inferir alguma relação entre grupos de neuroides [125]. É um tanto complicado em biologia detectar quais neurônios estão ou não participando de uma assembleia, porque isso

implica medir a correlação entre um grande número de unidades e de disparos. Santos e colegas [124], recentemente relatam o uso de *Principal Component Analysis* (PCA) na detecção do número de assembleias numa rede, da identificação dessas assembleias e da computação que a atividade das assembleias realiza em função do tempo. Embora o autor aponte que o método não seja apropriado (ou sensível) para detectar sequências de atividades neurais, como *synfire chains* (ver adiante), é importante observar trabalhos na direção oposta ao que se propõe aqui.

Os trabalhos desenvolvidos por Nicolelis e colaboradores (e.g. [127], [124]), tenta fazer uma engenharia reversa no cérebro, identificando assembleias que atuem na execução de funções motoras. Estes trabalhos podem fornecer vários *insights* para nossa abordagem, que é construir ou fazer o design e a concepção de assembleias dentro de redes neurais com o propósito de executarem operações computacionais. São, na verdade, abordagens complementares.

Há algumas décadas pouco se conseguia em termos de medição precisa e muito se inferia sobre formações de assembleia, embora tanto a formação de aglomerado de neurônios (*pools*) quanto a noção de coluna cortical tenham sido resultado direto de observação experimental [125].

A própria medição que um eletro-encefalograma (EEG) faz é resultado de aglomerados de padrões de disparos acontecendo ritmicamente. Mais recentemente, técnicas de gravação com micro-eletrodos, como as desenvolvidas pelo grupo de Nicolelis, têm esclarecido o assunto de formação de assembleias.

Pequenos grupos de neurônios pulsando conjuntamente, durante um intervalo de tempo restrito, têm sido descritos por termos e expressões como “população local oscilante” ou “oscilações síncronas” [128], ou “laços reverberantes” (*reverberating loops*) [129], ou ainda como “sincronização coletiva transiente” [77].

Nos últimos anos, o termo “assembleia” passou a ser mais regular na literatura científica [70], [124], [127], [130], remetendo principalmente à metafórica ideia de que “muitos neurônios agora estão de acordo”!

Ainda nos anos 1990, Abeles cunhou o termo *synfire chains* para descrever a noção de camadas de neuroides (*pools*) conectados em uma rede *feed-forward* disparando

em intervalos preditivos, com longos intervalos de atraso entre *spikes* [46], [44]. Segundo Abeles, numa tradução livre nossa,

uma *synfire chain* é essencialmente uma rede *feed-forward* de neurônios com muitas camadas (*pools*). Cada neurônio em um *pool* alimenta muitas conexões excitatórias no *pool* seguinte, e cada neurônio no *pool* receptor é excitado por muitos da camada anterior. Quando a atividade nesta cascata é organizada como um “bando” de *spikes* propagando sincronamente de camada para camada isto é chamado *synfire chain* [44].

A ideia de *synfire chains* remete a grupos de neurônios sincronizados. Outra possibilidade é que unidades formem grupos *multiplamente sincronizados*, ou que tenham pulsos “temporalmente atados” (*time-locked*), sem necessariamente estarem em sincronismo, como no conceito de “*synfire braid*” [45], [46].

Apesar do conceito “*assembleia neural*” ser difundido e discutido tanto em neurônios biológicos quanto em neuroides e redes artificiais, ainda não se sabe muito sobre os mecanismos pela qual neurônios retêm dados e organizam os processos computacionais por meio de formação de assembleias.

Segundo Paugam-Moisy e Bohte [22], “The current consensus agrees that cognitive processes are most likely based on the activation of transient assemblies of neurons [...], although the underlying mechanisms are not yet understood well.” A seguir, propomos um tipo de formação de coalizão, síncrona ou assíncrona que, acreditamos, seja um fenômeno fácil de acontecer em meio biológico dado a sua simplicidade e seja capaz de executar duas tarefas importantes em computação: reter dados biestáveis e organizar hierarquicamente processos computacionais. Antes, vamos agregar um conceito importante que usaremos adiante.

4.4.1 Grupos síncronos e policronizados (GP)

A questão de maior importância na formação de assembleias em RNPs é que os PAs ou *spikes* que atingem as sinapses de uma unidade cheguem dentro de janelas de tempo estritas. Em outras palavras, nas redes pulsadas as unidades podem agir como detectores de “coincidência”. Computação por assembleias é computação por coincidências.

Por isso a introdução de atrasos na rede de neuroides é uma necessidade, caso se pretenda simular toda a potencialidade das coalizões. Ao introduzir atrasos numa RNP, Izhikevich (ver [90] e [46]) mostrou que neuroides esparsamente conectados, com atrasos axo-dendríticos aleatórios, podem disparar seguindo padrões “temporalmente atados” (*time-locked pattern*).

Figura 4.2 – **Assembleias e grupos policronizados**. No topo é mostrada uma RNP, que pode ser uma rede recursiva (RNPR), um reservatório (*reservoir*), ou uma rede unidirecional (*feed-forward*). Nela, um núcleo (*kernel*) formado pelos neuroides P1, P2 e P3 detectam um padrão, e disparam em conjunto, ou seja, formam uma coalizão ou assembleia durante um breve intervalo. Os neuroides do *kernel* P estão conectados a outros neuroides, entre eles, A1, A2 e A3. O padrão (temporalmente atado) de disparo do *kernel* P e os atrasos na propagação até os neuroides do *kernel* A provocam em A1, A2 e A3 um padrão *policronizado* de disparo; ou seja, o núcleo A forma um grupo policronizado (GP). De forma semelhante, o padrão de disparo do *kernel* A e o atraso até o *kernel* B fazem com que os neuroides B também disparem como um GP. Mais ainda, o *kernel* B está totalmente conectado com A, e o padrão de disparo de B (mais os atrasos B-A) repetem em A o padrão de disparo iniciado pelo kernel P, criando um *loop* no qual a coalizão A dispara B, e B dispara A: dois GPs reverberando.

Tais padrões revelam que subconjuntos de neuroides podem ser ativados numa cadeia de reações [90], [46]. Como os padrões de disparo dos aglomerados de neuroides (*pool*) não são síncronos, apenas temporalmente conectados uns aos outros, Izhikevich chamou tais formações grupais de “*polychronous groups*”, que numa tradução livre seria “grupos policronais”. De acordo com o autor, “*policronal*”

deve ser distinguido de “assíncrono”, “parcialmente síncrono”, ou de “multiplamente sincronizado” que podemos chamar “polissincronizado” (ver [46], [47]).

Um *grupo policronizado*¹⁹ (GP) é caracterizado por um padrão de disparo temporal preciso em um *pool* de neuroides, que são mais prováveis de acontecerem do que simplesmente por acaso durante as atividades da rede.

Tais padrões acontecem principalmente por causa da topologia da rede, ou seja, da conectividade das unidades, dos pesos sinápticos e devido ao atraso das conduções dos pulsos nos percursos axo-dendríticos. A figura 4.2 ilustra o conceito de policronização. No topo da ilustração vê-se um conjunto neuroides de uma RNP que pode ser um reservatório (*reservoir*), uma rede pulsada recorrente, ou ainda uma rede *feed-forward* pulsada. Considere que os neuroide P1, P2 e P3 formem um núcleo (um *kernel*) e que tenham detectado um “padrão de disparo” no fluxo de pulsos que os estimulam. O padrão ativa primeiro P2, depois P1 e mais tarde P3.

Estes três neuroides podem estar conectados a milhares de outros neuroides, e certamente estão totalmente conectados aos três neuroides abaixo A1, A2 e A3.

Analisemos o que ocorre na propagação dos *spikes* que saem do *kernel* P. Um *spike* de P2 demora 9 ms para atingir o neuroide A1, depois leva mais 1 ms para o mesmo *spike* atingir A2, e ainda mais 1 ms para atingir A3.

Depois do disparo de P2 é a vez de P1 disparar com 2 ms de atraso em relação ao P2. O neuroide P1 dispara e demora 7ms para seu *spike* atingir A1, mais 1 ms para atingir A2 e mais 1 ms para esse mesmo *spike* atingir A3.

Continuando a sequência do padrão de disparo do *kernel* P, depois que P1 dispara é a vez de P3 disparar com 4 ms de atraso para P1. Porém, como P3 está mais “perto” de A1, A2 e A3, demora apenas 3 ms, 4 ms e 5 ms para que o *spike* de P3 atinja estes neuroides, respectivamente.

Dessa forma, temos uma *coincidência* acontecendo em cada um dos três neuroides do núcleo A (A1, A2 e A3). Os três pulsos do núcleo P (P2, P1 e P3) chegam

¹⁹ Preferiu-se o termo *policronizado*, em vez de *policronado*, por ser mais sugestivo em relação a grupos com uma relação temporal múltipla e travada, em certo grau, lembrando “sincronizados”.

exatamente no mesmo instante a A1, o que é suficiente para fazer este neuroide disparar. Apenas 1 ms depois os três *spikes* do núcleo P atingem o neuroide A2, e ele também dispara, mas, note-se que acontece 1 ms depois de A1. Então, 1 ms mais tarde, os três *spikes* do núcleo P coincidem novamente em A3, portanto, o neuroide A3 também irá disparar, 1 ms atrasado em relação a A2.

Sempre que certo padrão de pulsos acontece no fluxo de entrada da rede ilustrada no topo, o *kernel* P ativa sequencialmente os neurônios P2, P1 e P3, nesta ordem e com o mesmo intervalo de tempo de ativação entre eles. Ocorre que os neuroides do *kernel* P têm uma sequência temporal com atrasos de 2ms de P2 para P1 e 4ms deste para P3. Dessa forma, diz-se que o padrão detectado na rede provoca uma sequência temporalmente travada (*time-locked*) de disparos em P2, P1 e P3. Esse fenômeno de coesão temporal entre os pulsos dos grupos de neurônios/neuroides é denominado *policronização* (do grego, *poli* = múltiplos e *cronos* = tempo) [46]. Nenhum dos neuroides do grupo pulsa sincronamente, mas também não pulsam assincronamente, e nem estão multiplamente sincronizados, ou sincronizados com múltiplos sinais; eles pulsam deslocados no tempo.

Observando a figura 4.2 vemos que os neuroides P1, P2 e P3 formam um grupo policronizado (um GP) que ativa o outro núcleo formado por A1, A2 e A3; que também é um GP. Veja que o núcleo A, além de fazer conexões com muitos outros neuroides, está totalmente conectado ao núcleo B (neuroides B1, B2 e B3). Mais ainda, os neuroides do núcleo B também estão totalmente interconectados aos neuroides do grupo A, formando um laço de *conexões recursivas*. Como todos estes grupos pulsam policronizados, podemos dizer que o GP A está conectado ao GP B que, por sua vez, retroalimenta o GP A.

Assim, os núcleos A e B formam um pequeno aglomerado (*pool*) de neurônios recursivamente conectados. Este tipo de estrutura deve ser comum em circuitos reticulares biológicos nos quais os neurônios estão fortemente interconectados. Note-se que os neuroides do *kernel* P, do *kernel* A e do *kernel* B não estão exclusivamente conectados entre si, eles estão multiplamente conectados a quase todos os vizinhos. Entre eles, apenas a relação temporal entre os atrasos de seus pulsos e os pesos sinápticos que ajustaram entre si é que os colocam na condição

de grupos policronizados. Em face de um *evento* temporal particular (o disparo policronizado de P), uma cadeia de eventos policronizados acontecerá ao pool.

Este tipo de estrutura (*kernel* A e B interconectados) forma uma assembleia de neuroides que funciona como um GP operando de forma biestável, ou seja, um laço operacional que promove reverberação entre A e B a partir de um evento inicial que os dispara (*triggering event*).

4.4.2 Assembleias biestáveis (GPB)

Continuando a análise da propagação dos *spikes* sobre a rede, ilustrada na figura 4.2, assim que os neuroides do núcleo P ativam o núcleo A, os neuroides A1, A2 e A3 disparam com atrasos de 1 ms entre eles.

Os *spikes* demoram um tempo para atingir os neuroides B1, B2 e B3. Isso pode ser visto no gráfico da figura 4.3. Este tipo de gráfico é muito usado para análises em redes neurais pulsadas, chamado *raster plot*, um mapa temporal que ilustra os instantes em que os neurônios/neuroides sob análise dispararam. Na figura 4.3 podemos ver P1, P2 e P3 ao topo, e em baixo deles os núcleos A e B.

A sequência de disparos do núcleo A, associado aos atrasos de propagação das conexões com os neuroides do núcleo B, faz com que os *spikes* atinjam o núcleo B em um processo similar ao previamente descrito entre os núcleos P-A; ou seja: o *spike* de A1 leva 5 ms para atingir B1, este mesmo *spike* leva 7 ms para atingir B3, e 9 ms para atingir B2. Um *spike* de A2 leva 4 ms para atingir B1, mas um *spike* de A3 leva apenas 2 ms para atingir o mesmo neuroide B1.

Como resultado B1 disparará devido à coincidência dos *spikes* atingindo-o simultaneamente; e o mesmo fenômeno ocorrerá em B2 e B3, mas cada um em um tempo diferente causados pelo atraso nas propagações, como destacados na figura 4.3. Assim, os neuroides do núcleo B dispararão com um padrão travado de disparo que, associado ao retardo na propagação de seus pulsos até o núcleo A, gerará um padrão de disparo no *kernel* A similar ao do núcleo P, no começo do processo.

Esta é uma condição auto-sustentada: o núcleo A excitará o núcleo B que em seguida excitará o núcleo A. Esta é a condição em que dois núcleos (*kernels*)

formam um aglomerado (*pool*) operando de forma biestável, que será chamado aqui *Grupo Policronizado Biestável* (GPB). Em inglês, em trabalhos publicados ou enviados para periódicos, usamos o termo *Bistable Polychronous Groups* (BPG ou BPGs) [48], [49].

Figura 4.3 – **Raster Plot dos padrões de disparo dos GPs.** A relação temporalmente travada (*time-locked*) entre os três grupos policronizados pode ser vista neste gráfico. Em destaque estão os *spikes* do *kernel* A e B. O padrão de disparo do GP A associado aos retardos na propagação causam no GP B um padrão de disparo típico. Devido às interconexões entre B e A, associado ao respectivo retardo de sinais de B para A, o GP B causa no GP A um padrão de disparo idêntico ao que inicialmente foi causado pelo GP P. Cada grupo policronizado nesta rede é uma assembleia. Assim, a assembleia P dispara a assembleia A, que dispara a assembleia B, que forma um *loop* reverberante com A.

Os núcleos A e B são exemplos de ocorrências de GPs que, ao se ativarem mutuamente, formam um GPB ou uma assembleia biestável. Note-se que qualquer neuroide do núcleo A ou B poderá participar da formação de outros grupos policronados, ou seja, de outras coalizões. Contudo, um GP é tipicamente formado pelos mesmos neuroides e com as mesmas relações temporais entre as excitações e os disparos entre os neuroides participantes. Note-se ainda que a ordem de disparo dos neuroides (*ranking order*) é importante, e que se houver a possibilidade de núcleos dispararem em ordem diferente, apesar de conterem os mesmos neuroides, pode-se considerar que são GPs diferentes, ou seja, são coalizões diferentes dessa coalizão ilustrada acima.

Evidentemente, grupos com uma relação temporal síncrona também podem formar assembleias. Por exemplo, se todos os neuroides de P dispararem ao mesmo tempo, eles formam uma coalizão síncrona. Se os atrasos de P para A, forem todos iguais, A também disparará sincronamente, e formará uma assembleia síncrona; e se os atrasos de A para B forem iguais, igualmente B formará uma assembleia síncrona, que com atrasos idênticos para A, formará um *loop* reverberante síncrono. Note-se então que disparos de grupos síncronos é um caso particular de grupos policronizados, nos quais os tempos de disparo estão travados e são idênticos.

Conforme será discutido adiante, os GPs e os GPBs podem ter papel importante na computação em redes pulsantes. Primeiro, porque são agrupamentos que se formam espontaneamente (ver [91], [90]) e facilmente podem ser dissolvidos por inibição. Assim, um grupo policronizado biestável pode reter informação binária por tempo indefinido sem que mecanismos de plasticidade hebbiana fossem necessários, embora estejam envolvidos porque o uso de STDP na simulação causará o fortalecimento do GPB; ou seja, uma assembleia biestável consegue reter um bit de informação até ser dissolvida, e ainda que suas sinapses possam sofrer fortalecimento nos pesos, ela deixará de funcionar tão logo seja inibida. Segundo, tais assembleias podem reter estados operacionais e controlar fluxos de processamentos hierárquicos, discutidos no próximo capítulo desta tese.

4.5 Resumo do capítulo

Que tipo de computação um neurônio isolado é capaz de executar? Como vimos na figura 2.1, quando excitado, neurônios podem responder de forma linear aumentando a taxa de disparos em função da excitação de entrada, como os do tipo *class 1 excitable* (G). Outros podem oscilar e funcionar como osciladores intrínsecos, gerando *bursts* periódicos, como os do tipo *tonic burst*, figura 2.1(C). Este tipo de neurônio é chamado de *pacemaker* e é importante porque pode gerar sinais de sincronismo que pode favorecer a formação de coalizões. Mas, não há muito que se possa esperar de um neurônio isolado.

Dessa forma, esse capítulo se preocupou com o nível mais simples de conexão entre neurônios e neuroides, com as topologias que geram padrões de disparo ou CPGs, e ainda com o conceito de formação de assembleias.

Como o assunto do capítulo está um nível acima do nível de análise celular, primeiramente foi feita uma discussão sobre estratégias de simulação em redes neurais pulsadas. Mostramos que é possível criar redes *feed-forward*, redes pulsadas recorrentes (RNPR) e mais recentemente as redes baseadas em reservatório (*reservoir computing*).

Há duas estratégias básicas que podemos adotar para simulação de RNPs: baseadas nas iterações (clock-driven) e baseadas em eventos (event-driven). Nossa abordagem da computação por assembleias é essencialmente disparada por eventos. Assim, quando um grupo de neurônios dispara em conjunto consideramos que uma assembleia se formou, e esse é um *evento* importante para a rede.

Osciladores podem ser importantes para geração de sincronismo ou favorecerem o surgimento de assembleias, e vimos que um *pacemaker* pode ser um oscilador intrínseco. Porém, circuitos usando neurônios que se excitam ou inibem mutuamente têm sido relatados como circuitos básicos de geração de sinais de referência internos em animais. Um circuito que pode fazer isso é o mostrado na figura 4.1 que consiste de apenas dois neuroides.

Na natureza encontramos vários circuitos mais complexos que são capazes de gerar padrões de disparos igualmente complexos. Estes circuitos são chamados CPGs e no capítulo destacamos os motivos pelos quais não consideramos CPGs como coalizões, embora tecnicamente eles não apresentem diferenças, a não ser pelo fato de serem *hardwired*. Mostramos também que uma teoria defende que todos os circuitos do neocórtex advêm de circuitos de CPG, porém com maior plasticidade. Essa teoria vem ao encontro do que se pensa nesta tese: as assembleias funcionam muitas vezes como CPGs, porém, são plásticas e podem ser desarmadas facilmente.

Assembleias se formam espontaneamente porque neurônios/neuroides estão conectados uns aos outros, ou seja, por terem conexões sinápticas com pesos ajustados; mas tão influentes quanto os pesos sinápticos são os retardos nas

propagações de pulsos entre neurônios. Um tópico no capítulo analisa o aspecto de atrasos e frequências máximas que assembleias reverberantes podem apresentar.

É muito difícil detectar formações de assembleias em neurônios naturais, especialmente *in vivo*, dado o grande número de unidades e a explosão combinatória que estes fluxos de *spikes* propiciam. Recentemente, presenciamos um avanço nos dispositivos e nas técnicas de captação e gravação de neurônios individuais e em populações de neurônios.

Mas, é a aplicação de processos computacionais sobre esses dados (e.g. métodos como PCA) que permite identificar coalizões se formando, quais neurônios pertencem a quais assembleias e qual classe de operações funcionais eles estão executando. Contudo, ainda é difícil criar algoritmos que identifiquem sequências de assembleias, como as que propomos nesta tese, como explicam Santos e colaboradores [124]. Investigações neste sentido podem ser fonte de muitos *insights* para projetos de topologias de redes que operam como assembleias.

Depois das explicações preliminares que contextualizam o cenário como entendemos as assembleias, o capítulo explica o que são grupos policronizados, os GPs, como eles se formam e mantêm uma relação temporalmente travada (*time-locked*) entre os disparos de seus pulsos. Grupos sincronizados, como *synfire*, são casos particulares de GP.

A partir dos GPs mostramos por meio de um exemplo, na fig. 4.2 e o gráfico *raster plot* dos *spikes* da fig. 4.3, como acontece uma assembleia. Inicialmente, o grupo **P** forma uma coalizão e dispara uma sequência de pulsos que, dada a relação de atrasos com **A**, provoca a formação da assembleia **A** (fig. 4.2 e 4.3).

Uma das questões que temos que ter em mente é: se assembleias se formam, qual a razão de elas existirem por um brevíssimo intervalo de tempo e simplesmente desaparecerem? Só faz sentido pensar que as assembleias existam nos SNs se existir algum mecanismo que “detenha” essa informação, ou seja, que “registre” sua existência. Uma das hipóteses dessa tese é que *uma das formas de persistência do evento de uma coalizão é que ele reverbere em formações biestáveis*.

Da mesma forma, o GP **A** dispara e, dada a relação temporal com o grupo **B**, a assembleia **A** provoca o disparo de **B**. E como todo o grupo **B** está conectado ao

grupo **A**, formando uma retro-alimentação, o disparo da coalizão **B** associado à relação de atrasos de propagação que **B** mantém com **A**, provoca nos neuroides A1, A2 e A3 o mesmo padrão de disparo que o GP **P** inicialmente provocou.

Essa é a descrição de uma formação típica de assembleia reverberante, ou uma assembleia biestável, que é capaz de reter informação de um bit (na verdade, do evento **P** que a disparou). Denominamos esta formação típica de assembleias biestáveis de GPBs, ou grupos policronais biestáveis.

Este tipo de formação, neurônios amplamente conectados uns aos outros é uma constante em tecidos biológicos, especialmente em redes reticuladas e no neocórtex. Portanto, o fenômeno da reverberação ou da biestabilidade entre assembleias é totalmente plausível que aconteça em SNs biológicos.

Como a formação de coalizões, a dissociação delas e a reverberação entre GPBs podem *representar*, *controlar* fluxos algorítmicos e *organizar* hierarquicamente operações computacionais é assunto do próximo capítulo.

5 ALGORITMOS EM ASSEMBLEIAS NEURAIS

//

De acordo com D. Marr, máquinas computacionais, ou sistemas complexos de processamentos de informação, devem ser analisados ou entendidos em três níveis de processamentos: *implementação*, *algoritmo*, e *computação* [131].

No caso do processamento de informações em substratos neurais, no nível básico de *implementação* física, a preocupação do pesquisador deve girar em torno da biofísica dos neurônios, da plasticidade celular, da topologia da rede, da dinâmica das sinapses, do equilíbrio funcional na rede, do entendimento da rede como sistema dinâmico etc; assuntos dos capítulos anteriores.

No mais alto nível, o da *computação*, as preocupações devem se voltar para as teorias da *computação* e da *computabilidade*, que se preocupam formalmente com os tipos de problemas que podem ser resolvidos em um dado modelo de computação. Uma descrição sobre o que queremos dizer quando usamos o termo *computação* pode ser encontrada em Koch, tradução livre nossa: “qualquer processo físico que transforme variáveis pode ser pensado como uma computação, desde que ela possa ser mapeada dentro de uma ou mais operações matemáticas que realize alguma função.” [7]. O nível computacional envolve descrição da função matemática que o sistema pode executar; suas relações e suas propriedades. Trata-se do nível de maior abstração, uma vez que envolve expressar as leis que regem o sistema que executa a computação, ou as leis que descrevem as transformações que os sinais e as informações podem sofrer. Está fora do escopo dessa tese abordar teorias da computação aplicadas ao modelo aqui discutido.

Neste trabalho, a preocupação maior está em detalhar como grupos de neurônios pulsantes são capazes de executar algoritmos, como podem memorizar eventos e como conseguem representar objetos e abstrações na forma de conjunto de pulsos esparsos e/ou sincronizados. Em outras palavras, a abordagem aplicada aqui é de natureza “funcional” em vez de “formal”. Não obstante sua importância, uma incursão aplicando teorias computacionais à abordagem aqui apresentada ficará em aberto para futuros trabalhos.

Portanto, dos três níveis explicados por Marr, o nível intermediário associado ao nível básico é o que pauta o trabalho aqui apresentado. No nível intermediário estão os *algoritmos* usados para lidar com informações a fim de solucionar problemas, executar as funções, fazer chegar às soluções.

RELEVÂNCIA PARA A TESE:

A seguir será descrito como as assembleias podem executar *funções lógicas* no momento em que são formadas. Mais que isso, elas podem se ramificar e controlar o desarme de outras assembleias. Essa cadeia hierárquica de interações entre coalizões pode executar inúmeros tipos de processamento paralelo de eventos, de informações, além de executar o controle do processo.

Comparando com os níveis de investigação cerebrais discutidos no segundo capítulo, estamos agora investigando entre dois níveis: o nível dos subsistemas e dos aglomerados executando funções complexas e o nível do “comportamento social” das unidades, resultando numa computação da qual as próprias unidades não conseguem ter conhecimento do que executam. Por esse motivo, por não haver limites definidos entre os níveis, é que qualquer tipo de classificação toma um rumo mais didático que pragmático; contudo, usamos esta forma pictórica de níveis de investigação cerebral para situar o leitor sobre quais aspectos da investigação são abordados no capítulo.

Neste, neurônios surgem como elementos agrupados, alegoricamente, eles formam “coalizões sociais”. Desse agrupamento resultam pelo menos as seguintes operações computacionais: representação (de um evento), memorização (desse evento) e controle de um algoritmo (controle de uma sucessão de eventos causais que resultam em computação sofisticada).

Na maioria das vezes, não existe uma forma única de solucionar um problema computacional; assim, não existe um único algoritmo que traga “a” solução para um dado problema. Na natureza, é provável que incontáveis algoritmos tenham sido lentamente incorporados aos organismos, numa abordagem *bottom-up*, à medida que os níveis inferiores (as redes de neurônios) foram se formando. Sobreviveram aqueles animais com os algoritmos mais apropriados para lidar com informações circunstanciais em certos nichos, em certos ambientes, na forma de organismos mais adaptados.

Este capítulo está voltado para discussão sobre como algoritmos podem ser executados em RNPs, em especial por intermédio de assembleias neurais. Mais especificamente, discute-se aqui como assembleias neurais pulsadas síncronas e/ou policronizadas podem executar algoritmos que resultam na solução de problemas computacionais. Uma boa forma de dar uma ideia global sobre a execução de um algoritmo é por meio de um exemplo.

5.1 Ramificações e dissoluções de assembleias

Entendendo computação como qualquer processo físico capaz de realizar transformações em variáveis ou representações (de eventos internos ou externos), podemos pensar uma assembleia neural biestável, um GPB, como um elemento abstrato que acontece num meio físico (os neurônios pulsantes), que se torna operante durante intervalos de tempos específicos e que pode memorizar ou reter informação a respeito de um evento, de um acontecimento qualquer.

Vamos apresentar um exemplo para mostrar como assembleias podem computar: considere um agente (um robô, um organismo, um personagem virtual) controlado por uma RNP, capaz de tratar estímulos externos visuais e cheiro. Tais estímulos são ininterruptos, o que causa um fluxo constante de PAs/*spikes* de entrada (*streaming*). Durante este fluxo de PAs/*spikes* grupos de neurônios ou neuroides disparam coincidentemente em razão de estímulos específicos na entrada. Ou seja, grupos de neuroides são momentaneamente ativados na medida em que certos padrões de entrada acontecem e são detectados.

Por momentaneamente entenda-se ao mesmo tempo (sincronamente) ou travados no tempo (*time-locked*), dentro de uma janela temporal curta. Por exemplo, ao detectar padrões advindos de estímulo ligados a um cheiro específico, um grupo de unidades dispara em sequência travada, como o grupo policronizado (GP) **P** no exemplo das figuras 4.2 e 4.3 no capítulo anterior.

Considere também que, internamente, este agente é capaz de detectar baixo nível de energia e de gerar eventos quando isso acontece. Esse evento pode ser geração de trens de pulsos (*bursts*) ou a formação de uma assembleia. Vamos considerar a

última opção como exemplo. Isso significa que quando baixos níveis de energia são atingidos, grupos neurais geram eventos efêmeros que sinalizam, ou internamente *representam* baixos níveis energéticos para aquele agente. Assim, esse evento pode ser dito como representando “fome” para aquele agente; e nada mais é que uma assembleia neural pulsando a intervalos efêmeros e definidos. Para fazer algum sentido para a máquina computacional, tais eventos (detecção de cheiro, fome, etc.) devem disparar outros circuitos na RNP, alertando que uma solução é necessária.

Figura 5.1 **Ramificação de assembleias e eventos.** Na figura, vemos que o evento **FOME** (uma assembleia simples, um GP dispara dois GPBs (**A1-A2** e **F1-F2**) reverberantes, ou seja, biestáveis: **A1** dispara **A2** que retroalimenta **A1** e assim as duas se mantêm operacionais, idem para **F1-F2**). O GPB **A1-A2** é responsável pelo estado de **atenção** (visual e cheiro). O GPB **F1-F2** é responsável por controlar CPGs que fazem o agente *se mover* à procura de alimentos. A interação entre as coalizões resulta no processamento de informações e execução de tarefas (*algoritmos*). Interações entre GPs e GPBs podem disparar novos eventos, podem inibir assembleias ou executar funções lógicas, como a função **AND**: $(A1.A2)^1$, $(L2.L1.F2)$. Cada GPB, ao mesmo tempo que retém a informação (e.g. manter atenção, disparar forragear), também controla o fluxo hierárquico das ações paralelas a serem executadas; porque essas ações são disparadas pelas próprias GPBs. Portanto, os GPs e/ou GPBs internamente representam algo e, paralelamente, quase como consequência, controlam a hierarquia de um algoritmo, de uma sequência de ações que resultam um comportamento.

Se este agente for um organismo que sobreviveu e evoluiu, então, algum tipo de resposta comportamental foi sendo incorporada ao longo de sua evolução que lhe garante solucionar este problema. Ou seja, o organismo deve ter algum algoritmo que executa ações que lhe restabeleça os níveis energéticos. Na figura 5.1 vemos um hipotético algoritmo executado por algumas assembleias. No centro da figura

vemos o *evento* FOME pulsando periodicamente. O evento FOME pode ser a formação de um GP, uma assembleia síncrona, um neurônio isolado gerando rajadas periódicas. Ele é responsável por deflagrar uma série de eventos paralelos que se ramificam.

Considere que o algoritmo, extremamente simplificado, cuja finalidade seja resolver a reposição de energia do agente, possa ser descrito pela sequência de ações: “forragear”²⁰, manter “atenção” a sinais de alimento, “parar” quando encontrar algum sinal relevante, “deslocar-se” até a fonte; assim por diante, realizando os passos que solucionem o problema até que o estado interno “fome” não mais aconteça.

O que estamos propondo é que este algoritmo, de alguma forma, exista corporificado na máquina cerebral do agente e utilize o recurso das coalizões para executar as operações de representação e controle algorítmico. Em outras palavras, numa rede com milhares de unidades interconectadas existem várias conexões, vários pesos ajustados, e a serem ajustados, e várias relações temporais que permitem que os grupos síncronos ou policronizados aconteçam e interajam da forma descrita acima, executando estas sequências na medida em que as assembleias interajam entre si, ativando outros circuitos como CPGs etc. e, como resultado final, aconteça a solução do problema inicial: a motivação (o *driving force*) que impele o agente a responder da forma apresentada.

Esse algoritmo, em qualquer agente inteligente, deve ser adaptativo, porque, num meio em constante mudança, dificilmente ele se repetirá de forma idêntica, na mesma sequência, ou com a mesma temporalidade e intensidade.

Analisando o fragmento de algoritmo mostrado na figura 5.1 vemos que do evento **FOME**²¹ saem duas linhas com terminação de setas, denotando que o evento é excitatório; ou seja, os neuroides da assembleia **FOME** provocam *potenciais excitatórios pós-sinápticos* (PEPS) nos neuroides a eles conectados. Por outro lado,

²⁰ Forragear é um verbo que significa vaguear em busca de comida, procurar alimento. O termo inglês equivalente é *foraging* denota procura com afincos por alimento.

²¹ *Curiosidade*: nos mamíferos, um núcleo do hipotálamo (*hypothalamic arcuate nucleus* - ARC) é responsável por semelhante sinal de fome. Os neurônios são estimulados pelo peptídeo grelina disparado pelo estômago vazio na corrente sanguínea. Este núcleo é parte de um complexo sistema responsável pelo balanço de energia e controle da fome (ver [6] pp. 510-524, [142], [143], [144]).

as linhas terminadas com um círculo (que aparecem apenas na figura 5.2), denotam que os neuroides da assembleia provocam *potenciais inibitórios pós-sinápticos* (PIPS) nos neuroides da assembleia a eles conectados.

Em resumo, no gráfico da figura 5.1 as linhas terminadas em setas provocam disparo (*triggering*) de outras assembleias, enquanto as linhas terminadas em círculos provocam o desarme ou o silêncio (*inhibition*) de outras assembleias.

Note que no meio das linhas pode haver esferas representando grupos de interneurônios (figura 5.2). Eles podem ser facultativos em certas situações de formação de algoritmos e seu papel será discutido adiante.

Continuando a análise do algoritmo da figura 5.1, uma ocorrência do evento **FOME** é suficiente para disparar dois GPBs: a assembleia A_1-A_2 (que representa **ATENÇÃO**) e a assembleia F_1-F_2 (que representa **FORRAGEAR**). A assembleia **ATENÇÃO** estará ativa enquanto A_1 reverberar com A_2 ; da mesma forma, **FORRAGEAR** estará ativa enquanto F_1 reverberar com F_2 . O processo de uma assembleia (como **FOME**) disparar outras assembleias ou eventos é chamado aqui de *ramificação* (*branching*).

A ramificação pode ser entendida como o disparo de processos paralelos no algoritmo. Cada ramificação pode controlar um processo, ativar um estado interno, ou ligar/desligar um CPG, por exemplo. Pode ainda vetar outros processos, outros eventos ou outras assembleias, por um processo de inibição.

Assim, podemos observar que o GPB **ATENÇÃO** se ramifica em virtude da necessidade de controle de dois tipos de atenção diferentes: a atenção visual para objetos identificados como fontes de alimento e a atenção por cheiros conhecidos que indiquem alimentos por perto.

Consideremos então que a GPB **ATENÇÃO** controla o disparo de dois outros ramos da seguinte maneira: a coincidência temporal dos disparos de A_1 E de A_2 provoca o evento disparador (*triggering event*) de dois outros GPBs: uma assembleia biestável formada por S_1-S_2 (representando *atenção ao cheiro*) e outro GPB formado por V_1-V_2 (representando *atenção visual*). Note-se que é a função lógica **AND** (**E**) que está sendo executada.

Em outras palavras, a coincidência temporal de disparos de A_1 AND A_2 , denotado no gráfico por $(A_1.A_2)^2$, dispara o grupo policronizado S_1 . Na sequência, S_1 dispara S_2 que dispara S_1 novamente, tornando-se uma assembleia biestável, como é do conhecimento do leitor. Assim, o GPB S_1-S_2 coloca o agente em estado de atenção a qualquer padrão de cheiro de alimento, enquanto o GP V_1 reverberando com o GP V_2 (formando a GPB V_1-V_2) colocam o agente em estado de alerta para qualquer sinal visual de alimento.

Paralelo a esse processamento, o GPB **FORRAGEAR** dispara, também em duas novas ramificações, os GPBs **MV_PERNAS** e **MV_CABEÇA** (mover pernas e cabeça). Note que a ocorrência de F_1 E F_2 dispara tanto L_1 quanto H_1 ; embora a notação $(F_1.F_2)^1$ e $(F_1.F_2)^2$ no gráfico denotem dois eventos temporalmente distintos. Note que o evento neste contexto é sinônimo de coalizão ou assembléia. Por exemplo, da mesma forma que os anteriores, $(A_1.A_2)^1$ e $(A_1.A_2)^2$ são eventos diferentes, são assembleias que aconteceram em instantes diferentes, embora formadas pelos mesmos grupos de neuroides.

O resultado do acontecimento da coalizão $(F_1.F_2)^1$ é que esse evento dispara o GP L_1 , que reverbera com L_2 formando outro GPB. Continuando a ramificação num sentido *feed-forward*, a assembleia biestável L_1-L_2 gera eventos de controle para o CPG, por exemplo, que controla o movimento das pernas (não ilustrado), e o agente começa a se mover pelo ambiente.

Da mesma forma, a outra bifurcação gerada pela assembleia biestável F_1-F_2 começa na geração da coalizão $(F_1.F_2)^2$, evento este que dispara H_1 , que reverbera com H_2 formando a assembleia biestável (GPB H_1-H_2) que ativa o disparo dos CPGs que controlam o movimento da cabeça (também não ilustrado no desenho).

Assim como surgem ramificações, assembleias precisam ser desmanteladas; caso contrário a RNP continuaria executando o laço das assembleias biestáveis indefinidamente, levando toda a rede a uma atividade caótica, completamente ativa o tempo todo, parecido com estados epiléticos. Em outras palavras, *vetos* e *inibições* são tão importantes quanto o são as *ramificações* e os *disparos* de novos grupos neurais. É preciso que ocorra o desmantelamento das assembleias

disparadas no algoritmo acima, caso contrário, o agente entraria permanentemente em estado de *atenção* e continuaria eternamente *forrageando* e movendo as pernas.

A figura 5.2 mostra uma hipótese simples de como uma assembleia pode ser desmantelada. Considere que um grupo de neurônios no sistema sensorio de cheiro detecte um padrão “conhecido” de estímulos de entrada; ou seja, um padrão de cheiro de alimento. Esses neurônios formam uma assembleia, o evento P_{sm} , que nada mais é que uma coalizão similar à **FOME**, só que formada por unidades inibitórias (ou excitatórias ligadas a interneurônios inibitórios).

Figura 5.2 **Veto ou inibição de ramos**. O agente está “forrageando”, com os estados de atenção visual e atenção a cheiros ativados. Num certo instante, um evento de cheiro de alimento conhecido P_{sm} é percebido. A coincidência entre o disparo da assembleia $S1$ E P_{sm} , OU a coincidência de $S2$ e P_{sm} , (denotado $((S1+S2).P_{sm})$), gera um pulso inibitório que desarma a coalizão biestável $L1-L2$, ou seja, silencia o GPB $L1-L2$. Ou seja, na cascata, o agente deixa de ativar os CPGs que controlam as pernas porque no evento $(L1.L2.F2)$ tanto $L1$ quanto $L2$ estão silentes. Provavelmente o agente passa a executar outra sub-rotina do algoritmo, que pode ser descobrir a direção de onde vem o cheiro (não mostrado na ilustração).

Dessa forma, uma vez que a assembleia biestável $S1-S2$ está ativa e dispara de forma persistente, a combinação de $(S1$ OU $S2)$ coincidente com (E) o evento P_{sm} formam outro evento interno: $((S1+S2).P_{sm})$ ou $S1.P_{sm}+S2.P_{sm}$. Naturalmente, este evento em vez de ativar ou ramificar assembleias ele inibe o GP $L1$, dissolvendo a

GPB **MV_PERNAS**. Colocando de outra maneira, uma assembleia formada por $S_1.Psm$, ou uma outra assembleia formada por $S_2.Psm$, pode desarmar o GPB L_1-L_2 . Considerando a inibição, o veto, o desarme de assembleias biestáveis como sendo uma negação da construção dos ramos, podemos assumir que a função dos circuitos inibitórios equivale à função lógica *NOT*.

5.2 Elementos da computação por assembleias

Uma observação sobre a notação usada neste exemplo: a expressão $(F_1.F_2)^{1,2}$ não quer dizer que todos os *spikes* de F_1 E todos os *spikes* de F_2 devem coincidir, como é de se esperar para a função booleana E (*AND*). Na verdade, a expressão quer dizer: “quando um número suficiente de *spikes* de F_1 coincidirem (por isso *AND*) com um número suficiente de *spikes* de F_2 , os neuroides excitados a partir da formação dessa coincidência dispararam, formando um evento, uma assembleia”.

Por outro lado, a expressão (X_1+X_2) nesta tese quer dizer que um conjunto de *spikes* de um evento ou assembleia X_1 OU um conjunto de *spikes* de X_2 acontecendo em conjunto podem disparar um evento seguinte. Mais ainda, essa notação quer dizer também que um número suficiente de *spikes* X_1 OU X_2 , somados ou acontecendo em conjunto, serão suficientes para gerar um evento (uma coalizão). Note então que são funções *pseudo-booleanas*²², ou são funções booleanas associadas com certa incertezas, com fenômenos estatísticos.

5.2.1 Funções lógicas estocásticas

Estas notações usadas acima querem descrever um fenômeno efêmero e estocástico. Mais ainda, elas devem se alterar o tempo todo na medida em que a plasticidade neural e sináptica (PH, intrínseca, STDP) forem inseridas no modelo. No texto são utilizadas notações como se fossem funções booleanas para facilidade de

²² Na falta de uma formalização e de uma notação mais precisa, denominou-se aqui “funções pseudo-booleanas” as representações usadas no texto, porque não são funções lógicas booleanas reais. Por exemplo, todas as entradas de uma *AND* devem ser **1** para que a saída seja **1**. Isso não influencia na compreensão da descrição funcional da computação por assembleias.

escrita, porém, são pseudo-booleanas. Será um desafio realizar um tratamento matemático para formalizar o fenômeno das assembleias, sua ocorrência e dependência temporal, sua relação com os *spikes* e pesos sinápticos de vários vizinhos, e ainda as influências que sofrem em virtude dos vários tipos de plasticidade, ficando como proposta para trabalhos futuros.

Dessa forma, nesta tese vamos usar o recurso das notações pseudo-booleanas para que nos permitam vencer uma etapa nesta investigação. Tendo isso em mente, o evento $((S_1+S_2).P_{sm})$, que provoca inibição, se comporta como uma função *NÃO* associada à função *E* (*NAND*). Mas o mesmo evento pode ser denotado por $S_1.P_{sm}+S_2.P_{sm}$, que descreve uma soma de produtos que executa uma função inibidora; neste caso, uma função *OU* (*OR*) associada a um *NÃO*, ou seja, uma função *NOR*. Mais ainda, é possível que um evento isolado, como a assembleia P_{sm} , dissolva diretamente um GPB, como $L1-L2$ do exemplo acima. Neste caso, o evento se comporta como uma função booleana *NÃO*, ou seja, uma coalizão inibitória, ao executar um veto, se comporta como uma função *NOT*.

Fazendo um comparativo com os elementos de eletrônica digital e as funções lógicas necessárias para implantação de circuitos lógicos elementares, podemos dizer que, do exposto acima, todos os elementos necessários para executar lógica combinacional e sequencial podem ser encontrados nas assembleias: temos todas as funções pseudo-booleanas básicas (*AND*, *NAND*, *OR*, *NOR*, e *NOT*) que podem ser combinadas e que nas assembleias exercem as funções de ramificar ou dissolver novos ramos de coalizões.

Mais que isso, vimos que GPBs podem reverberar e memorizar estados internos enquanto a coalizão estiver ativa; portanto, essas coalizões reverberantes executam operações biestáveis, que em eletrônica digital podem ser comparadas às funções dos circuitos digitais flip-flops. Já demonstramos como assembleias biestáveis podem executar funções básicas e atuar como flip-flops, fazendo-as atuar como contadores binários ou como contadores acumuladores [48], que será discutido no capítulo seguinte.

Dessa forma, uma das contribuições dessa tese é fornecer aos engenheiros uma visão de como neuroides podem gerar redes que funcionem como circuitos digitais

que se ramificam, e aos cientistas de outras áreas uma visão de que não é um neurônio individual que executa funções lógicas, como queriam McCulloch e Pitts; mas sim as assembleias, que as executam muito bem. Melhor ainda que as funções booleanas, nas quais todas as entradas importam, as funções pseudo-booleanas possuem margem para falhas em alguns dos participantes das assembleias. Ou seja, temos no comportamento de uma população de neurônios uma implantação das funções lógicas básicas operando de forma estatística.

5.2.2 Hierarquias e relações causais

Existe uma relação causal na ramificação das assembleias. No exemplo, **FOME** é o evento, a assembleia que causa **ATENÇÃO** e **FORRAGEAR**. Em um algoritmo mais completo, possivelmente, várias outras ramificações. Por outro lado, **FORRAGEAR** é o que causa a ativação dos GPBs **MV_PERNAS** e **MV_CABEÇA**. Existe uma hierarquia nestas relações.

O GPB **FORRAGEAR** pode não ser o único a disparar uma hipotética coalizão que controla o movimento das pernas (**MV_PERNAS**), porque este movimento pode ser intencional, disparado por outros centros. Contudo, no exemplo, a assembleia foi disparada pelo evento $(F_1.F_2)^1$ a partir de **FORRAGEAR**.

Hierarquicamente, para este exemplo de algoritmo, o movimento das pernas está subordinado o GPB **F₁-F₂**, assim como as outras assembleias ramificadas e dissolvidas durante o processo estão ligadas de forma causal às assembleias e eventos que as originam e/ou destroem.

Como foi dito, uma formalização matemática mais apropriada precisa ser desenvolvida a fim de criar uma notação eficiente para descrever coalizões simples e assembleias biestáveis. Assim, adicionalmente à formalização das relações acima descritas, temos ainda que incluir que todas as funções da cadeia, de certa forma, estão hierarquizadas pelas relações causais de um evento (um GP ou GPB). Parece ser um trabalho monumental, e como também já foi dito, fica como proposta para trabalhos futuros

A abordagem nesta tese tem um viés funcional, o que não quer dizer superficial. Estamos analisando as possibilidades funcionais e possíveis problemas que, em primeira vista, podem surgir na computação por meio de assembleias. A notação a ser criada deve dar conta de juntar os elementos básicos (os tijolos – as funções pseudo-lógicas básicas que as assembleias executam), mais as relações de recorrência que os grupos policronizados biestáveis apresentam e ainda levar em conta todas as relações hierárquicas e causais, conforme descrito acima.

Em resumo, assembleias podem controlar algoritmos à medida que controlam a ramificação e a dissolução de outras assembleias. Para que uma ramificação aconteça, um evento excitatório deverá ser gerado, e isso acontece na medida em que os disparos de um grupo de neurônios coincidem temporalmente (somatória espacial e temporal) com os disparos de outros neurônios. Quando isso acontece, podemos observar que as funções pseudo-booleanas *AND* e *OR* são executadas na rede. Por outro lado, é possível que grupos de neurônios coincidam com outros e gerem eventos inibitórios; dessa forma, podemos observar que as funções *NAND*, *NOR*, ou *NOT* são executadas na rede.

A função de memória ou flip-flop é executada na rede assim que assembleias biestáveis se formam. As coalizões biestáveis, dando suporte e, associadas à ocorrência das outras funções pseudo-lógicas acima descritas, permitem incontáveis arranjos de ramificações e desmantelamento de assembleias.

Talvez essa seja uma das maiores contribuições científicas dessa tese: a noção de que elementos de lógica digital emergem quando assembleias se formam e interagem com outras assembleias. A identificação de que esses elementos pseudo-lógicos emergem das coalizões neurais permite pensar projetos de redes neurais pulsadas utilizando elementos de projetos de circuitos eletrônicos digitais sequenciais. Em nossa proposta, as pseudo-booleanas e assembleias biestáveis (os flip-flops neurais) são os elementos básicos usados pelas RNPs para executar qualquer máquina de estados digital, ou para montar qualquer algoritmo em circuitos digitais pulsados.

Com esse pensamento em mente, um engenheiro agora pode pensar em desenvolver circuitos neurais como se fossem máquinas de estados digitais, ou em

um cenário melhor ainda, máquinas de computação paralela, com cada ramificação sendo uma nova assembleia emergente de processos e atividades anteriores. As possibilidades são enormes.

5.3 Semântica e comportamento do exemplo

O exemplo descrito acima pode ser semanticamente interpretado como um algoritmo de computação paralela que trata o problema proposto: quando está com fome o agente dispara internamente o evento **FOME** que, em princípio, ativa dois estados internos paralelos: num ramo, a assembleia biestável que memoriza e controla seu estado de “atenção”, e no outro ramo, a assembleia biestável que o coloca no estado “forragear” (em inglês, *foraging* – andar a procura de comida).

A assembleia **ATENÇÃO** ativa dois circuitos internos: a atenção visual e olfativa para sinais de alimento. Ao mesmo tempo, a assembleia **FORRAGEAR** ativa dois novos subsistemas para controle das pernas e da cabeça. A fim de transformar esses dois estados internos em movimento, presumimos que essas assembleias ativam os CPGs respectivos que cuidam de gerar padrões de disparo para os músculos das pernas, pescoço, etc. Assim, o agente se coloca em movimento no ambiente, prestando atenção para sinais de comida.

Quando um cheiro de alimento chega até o agente, circuitos internos detectam uma classe de disparos e geram um evento, sinalizando que um “padrão” de cheiro de alimento conhecido foi detectado. Como a assembleia atenção para cheiros de alimentos está ativa, a coincidência desses disparos provoca um evento que inibe o movimento das pernas. A execução de outras instâncias desse algoritmo certamente continuaria; por exemplo, a detecção da direção do cheiro, um movimento padrão nas pernas para mudança de direção, encaminhar-se no sentido da fonte, etc; nenhuma dessas instâncias é mostrada no desenho.

O que se quer apontar é que, *qualquer descrição do comportamento do agente, ou seja, do algoritmo que o agente está executando, pode ser transcrito para uma representação de assembleias neurais executando as pseudo-booleanas associadas às assembleias biestáveis.*

Naturalmente, a topologia da rede e a forma como as assembleias devem executar algoritmos estará diretamente ligada ao sistema sensorial do agente, e aos atuadores que o agente possui. Isso significa que as assembleias devem executar algoritmos em função dos sensores e atuadores corporificados ao agente (isso aproxima nossa abordagem das teorias e dos conceitos de *embodiement*²³).

Vaguear, ou mover-se, significa gerar padrões de disparos neurais de tal forma que alguns músculos são ativados enquanto outros são bloqueados. Assim, deslocar-se no meio ambiente significa gerar saídas comportamentais (ativar uma série de músculos) de forma apropriada. Isso significa que esta abordagem se complementa na (ou se aproxima da) proposta de que circuitos no *neocórtex* evoluíram a partir de circuitos rítmicos motores e de CPGs que originalmente se encontravam na medula espinhal e tronco encefálico, descrita por Yuste e colaboradores [120]. Em nossa visão, até certo ponto, o neocórtex reproduz em assembleias plásticas e muito mais numerosas, aquilo que CPGs fixos executam em circuitos mais rígidos.

Portanto, conhecendo-se os mecanismos que um agente usa para gerar ações, conhecendo-se o padrão de disparos que provoca cada ação, conhecendo-se a sequência de ações que geram dado comportamento e conhecendo-se que certos comportamentos devem ser aplicados na solução de certo problema, é possível partir de uma descrição semântica de um comportamento para executar um projeto de engenharia que solucione o problema utilizando RNPs e a abordagem das assembleias.

Em outras palavras, por mais complexo que seja o “algoritmo” que se deseja que agentes executem, é possível desenhar CPGs que executem as tarefas sequenciais dos efetores; depois projetar redes que disparem tais CPGs e, acima desse nível, projetar redes que executem funções de controle em assembleias biestáveis,

²³ *Embodied intelligence* ou *cognition*, ou *situated cognition*, e termos correlatos são conceitos que defendem que inteligência/cognição estão distribuídas pelo corpo do agente (corporificadas). Tais teorias afirmam que comportamento inteligente/cognitivo emerge da interação entre cérebros, corpos e mundo. Assim, um agente é capaz de aprender certas habilidades em virtude de possuir o aparelho adequado; ou seja, porque este agente possui a rede neural que processa informações e os atuadores que efetuam as ações, resultando num comportamento modificador do meio (ver [137], o conceito *enaction* (J. Brunner), e *situated / embodied agents* em [145]).

partindo das respostas comportamentais desejadas, das formas de entradas sensoriais do agente e dos atuadores que o agente possui.

5.3.1 Representação interna

Devemos notar que foram utilizadas palavras no texto acima para denominar certos *estados internos* do algoritmo exemplificado. Por exemplo, quando o agente atinge baixo nível de energia uma assembleia pulsa indicando, ou *representando* o estado interno **FOME**. Além disso, essa assembleia dispara ramificações colocando o agente em estados como **FORRAGEAR** ou **ATENÇÃO**.

Em outras palavras, as assembleias além de *controlar* hierarquicamente o disparo ou a inibição de outros processos do algoritmo podem *representar* estados internos. Nesse sentido, representar e controlar são a mesma coisa ao mesmo tempo. As assembleias **FORRAGEAR** e **ATENÇÃO** são biestáveis e podem *memorizar* informação: uma memoriza que o agente deve caminhar até encontrar alimento, enquanto a outra memoriza que ele deve estar atento por alimento. E enquanto memorizam tais estados internos elas controlam que outras ações devem acontecer.

Por outro lado, no algoritmo, quando um cheiro é percebido, outra assembleia dispara assim que o padrão conhecido é detectado. Isso quer dizer que as assembleias podem também *representar objetos e ações* que ocorrem *externamente* ao agente. Essa representação pode ser internamente memorizada na forma de assembleias biestáveis, em curto espaço de tempo; ou podem se tornar memória de longo prazo caso as regras de plasticidades aplicadas à rede fortaleçam as conexões daquela assembleia especificamente. Ou seja, as coalizões temporárias, ou não, podem explicar como um agente representa e memoriza internamente eventos do mundo exterior, para compará-los com padrões conhecidos.

Mais que isso, a própria comparação (ou tomada de decisão) pode ser executada no momento em que as assembleias executam as pseudo-booleanas, como é o caso do exemplo da detecção do cheiro $S_1.P_{sm} + S_2.P_{sm}$ no exemplo da figura 5.2. Para introduzir um elemento de reconhecimento interno, basta que em qualquer dos produtos entre um componente como $S_1.R_{ix}.P_{sm}$, com R_{ix} denotando qualquer padrão

de disparo interno que represente um cheiro específico. Mas o cheiro também pode estar internamente representado nos pesos sinápticos, em memória de longo prazo que impulsionam a geração do evento *Psm*. As possibilidades são muitas e, na medida em que novos avanços na neurociência revelarem novos mecanismos de memória, o modelo que apresentamos parece estar apto a testar tais possibilidades. Este assunto parece ser uma fonte promissora para futuras pesquisas.

Portanto, defende-se neste trabalho que assembleias biestáveis podem tanto *significar* (representar algo) quanto *organizar* complexos algoritmos computacionais paralelos, o que resulta nas mais complexas operações. Em vez de células representando conceitos, defendemos que as assembleias representam conceitos de forma esparsa, uma espécie de “*grandmother neural groups*” [12], [14], [22].

A execução de algoritmos depende da interação entre assembleias; ou seja, da recíproca e hierárquica ação de uma assembleia sobre outra. Certamente, não é tão simples assim obter computação por meio de assembleias e, numa primeira análise, há uma série de problemas na abordagem de computação por assembleias.

Tais problemas são os assuntos dos próximos tópicos neste capítulo. Além do mais, é possível que coalizões neurais não sejam a única forma de computação encontrada pela natureza. É provável que assembleias se associem a outras formas de computação neural, inclusive analógicas, utilizando o vasto repertório de comportamentos neurais apresentados no capítulo II; ampliando a capacidade computacional e, ao mesmo tempo, elevando o grau de complexidade e de compreensão do funcionamento desses circuitos. O recorte aqui feito, e a aposta em assembleias, é que podemos agora compreender como *representar*, *memorizar* e *controlar* algoritmos no mesmo substrato e na mesma forma operacional.

5.4 Questões sobre computação por assembleias

Há alguns pontos-chave em computação por meio de assembleias que serão discutidos a seguir. É importante salientar que, para gerar um *spike*, cada neurônio ou neuroide na assembleia deve receber um número suficiente de PEPS ou PIPS simultâneos (integração espacial) ou numa janela de tempo curta (coincidência

temporal). Estes pulsos advêm de eventos (neurônios disparando rajadas de pulsos, outras assembleias, etc). Assim, uma primeira questão a ser levantada está relacionada com os eventos recorrentes. E se os neurônios de uma assembleia receberem eventos de entrada de forma recorrente?

Outra questão está relacionada com o comportamento dinâmico de uma assembleia: por que assembleias binárias em vez de coalizões ternárias ou mesmo coalizões de múltiplos estados?

Uma terceira questão está relacionada com deslocamentos na temporização de eventos repetitivos em relação a um sinal interno. Temporalidade é importante para disparar, ramificar ou inibir assembleias; portanto, como as assembleias lidam com deslocamentos na fase dos eventos que as disparam?

RELEVÂNCIA PARA A TESE:

Os próximos tópicos tentam alertar para problemas que devem ser encarados na abordagem de computação por assembleias pulsantes. Os problemas têm a ver com estabilidade das redes de assembleias, com *phase-shift* nos sinais de entrada, e com recorrência de estímulos, que podem causar problemas similares aos encontrados em Redes Neurais de Pulso Acoplado (RNPA).

5.4.1 Assembleias multiestáveis

Vamos abordar primeiro a última questão colocada acima, a questão sobre múltipla estabilidade em assembleias, ou sobre múltiplos estados que uma assembleia pode assumir. Por que não abordamos até o momento assembleias com múltiplos estados? Pode haver assembleias com mais de dois estados? O que há de especial nas coalizões serem biestáveis?

Respostas para as perguntas podem requerer dois pontos de vista distintos: um abordando o *número de GPs ativos* dentro um *loop* biestável e outro abordando o *número de estados operacionais* que uma assembleia pode assumir.

Discutiremos este último ponto de vista agora e retornaremos ao outro em seguida. Considere cada padrão estável de disparo da assembleia um “*atrator*” na rede. É possível que assembleias reverberem em mais de um estado estável, ou seja, uma

assembleia pode reverberar em frequências diferentes. Consideremos este fenômeno como “*multiestabilidade*” na rede, seguindo [74]. Dessa forma, cada padrão de disparo distinto e razoavelmente separado dos outros, qualitativamente, pode ser considerado um *atrator* diferente. Neste sentido, assembleias multiestáveis são circuitos pulsados que possuem múltiplos atratores, ou seja, que podem apresentar múltiplos estados operacionais.

Por exemplo, considerando a frequência de operação da rede como um fator qualitativo que define atratores, um circuito neural poderia reverberar em 4 Hz (o primeiro atrator), ou em 40 Hz (segundo atrator), ou ainda em alta frequência, digamos 80 Hz (o terceiro atrator). Estamos retratando um circuito neural que, dependendo das condições de excitação, é capaz de mudar sua frequência de reverberação. Este exemplo descreve um sistema dinâmico ternário, com três atratores (4 Hz, 40 Hz e 80 Hz), em vez de apenas dois estados operacionais booleanos: *ativa* ou *inativa*. Num sistema com três atratores, a lógica deveria lidar com quatro estados (considerando o estado inativo: 0, 4, 40 e 80 Hz).

É possível que redes neurais orgânicas usem tal estratégia. Em SNs neurônios são sujeitos a mecanismos de plasticidade durante o tempo de vida do agente, como vimos no tópico 3.6. De acordo com Destexhe e Marder [40], neurônios individuais, sinapses e os circuitos que eles formam passam por mudanças de curto prazo assim como transformações de longo prazo, como resultado das experiências pelas quais os animais passam durante seu desenvolvimento, aprendizado e durante o presente comportamento. Neurônios naturais podem mudar sua própria excitabilidade, o que resulta em mudanças em seus comportamentos. Neurônios também modificam a condutância e o fator de acoplamento de suas conectividades sinápticas, o que pode resultar em mudanças na conectividade e no comportamento global de toda a rede. É importante levar em conta que, como discutido no tópico 3.6, sinapses morrem e novas são criadas durante o tempo de vida de um animal [103], o que dá ao animal a chance de adaptar tempos de propagação entre neurônios. Este aprendizado temporal é fundamental na formação de assembleias neurais, como vimos no tópico 3.7.1 (ver [106], [22]).

Em suma, mudanças plásticas nos neurônios e nas sinapses, associadas à natureza dinâmica das redes pulsadas, podem fazer surgir redes neurais biológicas com

múltiplos atratores, que operam em múltiplos estados estáveis. Múltipla estabilidade é encontrada em CPGs e até mesmo em neurônios marca-passo isolados [113]; portanto, é plausível acreditar que assembleias neurais também tirem proveito em ter um único circuito com múltiplos comportamentos.

Todavia, do ponto de vista computacional, é consideravelmente difícil lidar com circuitos lógicos com três estados. Apesar de lógica ternária ter aplicações²⁴ e possivelmente vantagens sobre a lógica binária, a implantação prática e comercial dos circuitos eletrônicos e computadores ternários ainda não se tornou realidade²⁵.

Assim, com três estados operacionais (*falso*, *verdadeiro* e *nulo*), aumenta-se consideravelmente a dificuldade dos circuitos lógicos eletrônicos. Nas RNPs seríamos obrigados a tratar três frequências diferentes na rede. Investigações em RNPs operando em N-estados com N-atratores são impraticáveis no presente momento. É aconselhável conhecer as propriedades e dinâmicas das assembleias biestáveis num primeiro momento, depois, se for o caso, explorar lógicas de múltiplos estados em redes com múltiplos atratores.

Até o momento, consideramos somente dois estados (*ativa* ou *inativa*) nas investigações com assembleias biestáveis (GPBs), e a frequência operacional é escolhida (um atrator simples) para a rede operar no estado *ativa*. Esta frequência é definida durante o projeto, determinada principalmente pelo atraso de propagação (*propagation delay*) dos *spikes* entre neuroides.

O estado inativo não quer dizer que o neuroide esteja totalmente silente, mesmo porque a plasticidade homeostática força as unidades a estarem operantes; dessa forma, um neuroide gerando pulsos em baixa frequência para se manter ativo não significa que a assembleia está ativa. Em outras palavras, o estado inativo de uma assembleia quer dizer que o evento não está acontecendo e não que os neurônios ou neuroides que delas participam estejam inativos. Até porque, neurônios e neuroides podem participar de várias assembleias diferentes. Múltiplos estados

²⁴ Por exemplo, em banco de dados como SQL que lidam com estado NULO.

²⁵ Podemos citar a tentativa russa de criação dos computadores "Setun" e "Setun 70" na Univ. de Moscou (ver <http://www.computer-museum.ru/english/setun.htm>). Acessado em 14.07.2011.

operacionais e múltiplos atratores em GPBs permanecerão como uma questão a ser investigada em trabalhos futuros.

Vamos então discutir o segundo ponto de vista sobre múltiplos estados nas assembleias, citado anteriormente. É possível cogitar que uma assembleia tenha mais de um estado quando leva em conta o número de GPs ativos para formar uma única assembleia biestável. Este assunto poderá ser mais facilmente entendido na forma de um novo exemplo.

Este assunto diz respeito também à formação de assembleias complexas, que envolve laços biestáveis mais difíceis de serem detectados ou construídos.

Figura 5.3: **Assembleias biestáveis complexas.** A figura mostra uma assembleia biestável com várias coalizões. O evento T_1 dispara dois GPs: a assembleia biestável D_1 - D_2 e a assembleia C_1 . C_1 dispara C_2 , que dispara A_1 , que dispara A_2 , e o GP A_2 dispara B_1 que dispara B_2 . Nenhum dos pares de assembleias C_1 e C_2 , A_1 e A_2 , e B_1 e B_2 possui loop biestável. O GP B_1 e D_2 e D_1 é que voltam a forçar o disparo da assembleia C_1 , fechando o loop biestável de todo o grupo.

A figura 5.3 mostra a formação de uma assembleia biestável complexa, na qual vários GPs estão envolvidos num *loop* operacional biestável singular. Vários GPs disparam uns aos outros formando uma cadeia que, ao final, trata-se de um único grupo policronizado biestável. O gráfico mostra uma assembleia formada por C_1 e C_2 . O GPB C_1 - C_2 não pode reverberar diretamente porque não há laço de *feedback* direto de C_2 para C_1 . Observe que inicialmente T_1 dispara C_1 que, por sua vez, dispara C_2 . Contudo, C_2 não retro-realimenta C_1 , pelo menos diretamente.

Na verdade, C_2 realimenta C_1 por meio de duas ramificações: C_2 dispara A_1 que dispara A_2 , mas A_2 também não realimenta A_1 . Em vez disso, A_2 dispara B_1 . A

assembleia **B1** dispara **B2**, e ao mesmo tempo seu disparo é uma das duas condições necessárias para fechar o *loop* com **C1**.

A outra condição advém da coincidência de disparos com os GPs **D1 OU D2**, denotado por $((D1+D2).B1)$. Este evento é o que dispara novamente **C1**, fechando o *loop* que torna este grupo de GPs numa assembleia biestável singular.

Este exemplo mostra quão complexa uma assembleia biestável pode ser. Embora seja possível argumentar que há muitos GPs assumindo diferentes estados neste *loop*, funcionalmente, todos os GPs e suas relações são consideradas como uma simples assembleia biestável que pode assumir o estado **ligada** ou **desligada**.

Este laço complexo é considerado uma única assembleia biestável do ponto de vista dos GPs **C1**; afinal **C1** voltará a ficar ativo ou não; ou seja, toda a assembleia está ou **ativa** ou **inativa**. Esta será outra dificuldade a ser abordada quando da formalização da computação por assembleias.

A abordagem de assembleias *com dois estados apenas*, ativas ou inativas, facilita o design das interações entre uma assembleia, sem que tenhamos que pensar em redes operando em múltiplas frequências. Portanto, podemos usar conceitos booleanos.

Além do mais, GPs são considerados ativos se estiverem reverberando em uma frequência estável (1), ou não estarão ativos (0) caso contrário, ou seja, se não houver reverberação alguma. Vale repetir que isso não tem nada a ver com uma unidade de um GP estar ou não disparando. Temos que pensar que neurônios e neuroides podem não disparar para uma assembleia específica, embora estejam ativos participando de outras coalizões e de outros processos que estão ocorrendo em paralelo na rede.

Portanto consideramos fora do escopo desta tese investigações sobre assembleias com operações *multiestáveis*, tanto do ponto de vista das operações em múltiplas frequências (com múltiplos atratores), quanto pela consideração das ramificações numa assembleia complexa como sendo estados múltiplos da rede.

Pelo menos neste estágio inicial da pesquisa, as investigações continuarão em torno de formações biestáveis, provavelmente não complexas que envolvam laços muito

grandes, para que possamos aplicar ao máximo os conceitos da lógica booleana e eletrônica digital em projetos de assembleias com RNPs.

5.4.2 Recorrência dos eventos de disparo

Considere o exemplo apresentado na figura 5.4, no qual dois neuroides formam um oscilador acoplado por pulsos.

Figura 5.4 **Influência de disparos recorrentes.** O circuito formado pelos neuroides **a** e **b** oscilam em uma frequência estável determinada por $\delta_{ba} + \delta_{ab}$. Ambos os neuroides tem uma entrada para corrente de excitação externa (I_a , I_b). O neuroide **a** possui um pulso excitatório acoplado Cp_a . O neuroide **b** possui um pulso inibitório acoplado. Assim, este circuito forma um PCO (*pulse-coupled oscillator*). Um pulso externo excitatório no neuroide **a** pode mudar a temporização de disparo de **a**, adiantando o tempo δ_a para δ_a' . Por outro lado, um pulso inibitório no neuroide **b** pode atrasar o tempo de disparo de **b**, aumentando o período δ_{ba} para δ_{ba}' . Ou seja, pulsos recorrentes numa assembleia reverberante modificam o padrão de resposta da rede, podendo colocá-la em regimes imprevisíveis ou até mesmo caóticos.

O neuroide **A** é disparado externamente, por meio da entrada Cp_A (excitatória), e inicializa uma operação biestável: **A** dispara **B** depois de δ_{ab} ms, e **B** dispara **A**

depois de δba ms. Assim, em regime estabilizado e sem influência externa, o circuito reverbera numa frequência constante.

A frequência de operação pode ser calculada por:

$$F = 1/(\delta ab + \delta ba)$$

onde δab denota o retardo na propagação de A para B , enquanto δba denota o atraso de B para A .

Observe que o neuroide B tem uma entrada de estímulo externo Cp_B , que é uma entrada inibitória. Devemos então analisar o que acontece em dois casos: o que acontece se A receber um estímulo excitatório durante a operação ou B receber um estímulo inibidor?

Em regime estável, um pulso externo excitatório em A pode provocar uma mudança, em geral um adiantamento do próximo pulso em A em relação ao regime normal; ou seja, geralmente ocorre uma diminuição de δba , denotado por $\delta ba'$ na figura.

Por outro lado, um pulso inibitório em B pode fazer com que o neuroide B leve mais tempo para disparar; ou seja, geralmente acontece um atraso do pulso de B em relação ao tempo que ocorreria em regime normal; ou seja, geralmente ocorre um incremento no tempo δab , denotado por $\delta ba'$ na figura. Em qualquer dos casos, a frequência do oscilador será alterada por meio da interferência externa.

Trabalhando com neuroides como o I&F, ou seja, com integradores simples, estes dois tipos de respostas podem ser esperados e o sistema se comporta de forma previsível. Contudo, “*neurônios são sistemas dinâmicos*, portanto eles devem ser estudados como tal” [25], então as respostas desse pequeno sistema acoplado não são tão previsíveis assim. Dependendo do modelo de neuroide escolhido para o circuito, influências externas podem provocar grandes alterações na estabilidade de um circuito biestável oscilante, como o do circuito acima.

O comportamento descrito neste exemplo caracteriza um oscilador acoplado por pulsos, ou uma Rede Neural de Pulso Acoplado (RNPA, em inglês: PCNN – *Pulse-Coupled Neural Network*). O raciocínio aplicado sobre um laço com apenas dois neuroides é válido também para assembleias neurais com vários neuroides.

Descrições formais dos comportamentos de redes de pulso acoplado foram desenvolvidas por Canavier & Achuthan, num trabalho que descreve desde laços com dois neuroides até redes com neuroides totalmente interconectados [114] (ver também [122]).

É possível que tal formulação possa ajudar a predizer o comportamento das assembleias, embora isso não tenha sido tentado. Também é possível que uma rede RNPA apresente multiestabilidade, como foi visto no tópico anterior. O maior problema com assembleias recorrentemente excitadas é que elas podem modificar a frequência de reverberação para qualquer harmônico, ou mesmo trabalharem na frequência máxima. Esse tipo de comportamento é descrito em situações de epilepsia em redes neurais naturais.

Utilizando um exemplo numérico, digamos que os dois neuroides formem um oscilador cuja frequência seja 25 Hz. Isso significa que 40 ms é o tempo total envolvido na propagação dos pulsos ($\delta_{ab} + \delta_{ba}$). Fazendo os atrasos iguais, temos $\delta_{ab} = \delta_{ba} = 20$ ms, em regime estável.

O problema é que o neuroide *A* deve ser estimulado externamente para que a assembleia inicie operação; assim como acontece com os GPBs *A*₁-*A*₂ e *F*₁-*F*₂ no exemplo das figuras 5.1 e 5.2, que são disparados pelo evento **FOME**.

Isso significa que o neuroide *A* do circuito acima está sujeito a receber um novo pulso a qualquer momento no intervalo de tempo em que está silente. Se ele receber um pulso, digamos 10 ms após o primeiro disparado, e esse pulso o fizer disparar novamente, este novo pulso viajará até o neuroide *B*. Como os tempos de propagação são constantes, um pulso surge em *B* 10 ms após o primeiro. E este novo pulso também fará *B* disparar novamente, pulso este que será refletido para *A*; ou seja, a frequência de reverberação do laço *A-B* dobra. Se o neuroide *A* continuar recebendo pulsos externos em tempos aleatórios, o laço *A-B* tenderá a oscilar na frequência máxima que os neuroides respondem, ou a se comportar de forma

caótica, ou ainda a parar de oscilar, dependendo do comportamento dinâmico dos neuroides²⁶ escolhidos para o circuito.

Em outras palavras, caso o pulso excitatório externo acoplado ao oscilador aconteça em um ritmo igual a um dos harmônicos, como o segundo harmônico introduzido a 10 ms no exemplo acima, a rede sincroniza numa nova frequência; contudo, caso o pulso externo não seja tão regular (mais provável em sistemas biológicos), então a rede tenderia a se comportar de forma imprevisível.

Fenômeno contrário ocorre para o caso do neurônio **B**: se ele receber forte estímulo inibitório repetitivamente, digamos os mesmos 10 ms, ele poderá não disparar nunca, e a assembléia não se formará. Ou seja, se **A** estiver recebendo estímulos excitatórios contínuos e **B** também estiver sendo inibido continuamente pode ocorrer de o estímulo contínuo de **B** não permitir que a assembleia se forme, embora **A** dispare constantemente.

As possibilidades de acontecer qualquer estágio intermediário imprevisível são enormes; ou seja, a natureza deve ter encontrado um meio de tornar o sistema robusto o bastante de forma que, se houver computação por assembleias biestáveis nos SNs, de alguma forma elas não devem sofrer influências recorrentes.

Uma possibilidade de tratar eventos externos recorrentes de maneira simples é bloquear os pulsos de disparo tão logo uma assembleia se torne ativa, ou seja, desacoplar a fonte de pulsos recorrentes.

Trata-se de uma estratégia mais apropriada para esta fase de investigações com GPBs, sem ter que considerar uma abordagem à base das RNPA's, que são bastante complexas em se tratando de redes cujos componentes (neurônios e neuroides) são sistemas dinâmicos não-lineares.

Dois motivos nos fazem crer que este é um caminho adequado: primeiro, uma assembleia biestável não é, por ela mesma, um oscilador. Ela forma um sistema que reverbera em razão de se tornar ativa, e não que os neurônios sejam sistemas oscilantes naturais acoplados a uma fonte de pulsos que trabalha corrigindo ou

²⁶ Alguns neurônios podem deixar de operar em certas frequências de excitação, por exemplo, neurônios com latência de disparo ou ressonadores [26], [25].

influenciando a próxima fase de disparo do oscilador. Segundo, não é difícil inibir uma fonte externa de pulsos enquanto a assembleia está reverberando, assim, é possível que sistemas naturais tenham escolhido este caminho também.

Uma solução para bloquear pulsos recorrentes pode ser inserir inter-neurônios/neuroides que funcionem como mediadores entre a fonte excitante e as unidades da assembleia. Neste caso, as unidades intermediárias podem agir como “portas” lógicas que são inibidas assim que as assembleias se tornem biestáveis.

Outra solução pode ser o auto-bloqueio dos pulsos recorrentes, por exemplo, com os neurônios/neuroides bloqueando as próprias entradas que os ativaram; contudo, esta é uma solução que requer respaldo e verificação empírica para se verificar se um processo dessa natureza ocorre ou não em redes neurais biológicas²⁷.

Certamente a natureza pode ter encontrado outras formas para solucionar o problema da recorrência, e este é outro assunto promissor para futuras investigações. Parece que duas estratégias principais podem solucionar o problema de forma satisfatória: uma, a da inibição da fonte recorrente, e outra, a da adaptação do comportamento da rede, operando como uma rede acoplada, de forma que a frequência da assembleia biestável se adapte à frequência de excitação, ou a um de seus harmônicos; ou seja, fazer a rede operar em um dos *atratores* naturais, oscilando em sincronismo na frequência de um dos harmônicos naturais.

5.4.3 Deslocamento de fase

Considere o exemplo do tópico 5.1 no início do capítulo (figuras 5.1 e 5.2) e um algoritmo simulando aquela rede, em tempos discretos com 1 ms de resolução. Observe que na figura 5.2 o evento *Psm* acontece em virtude de um estímulo externo. Dessa forma, ele pode ocorrer a qualquer momento, e isso é um problema para a abordagem apresentada aqui, uma vez que um deslocamento de fase em um evento de disparo pode romper todas as relações temporais entre assembleias.

²⁷ Teoricamente, neurônios podem auto-bloquear ramos de sua árvore dendrítica (ver *computação dendrítica*, tópico 3.7, fig. 3.4); porém, até onde foi possível pesquisar, não encontramos descrição desse fenômeno em neurônios biológicos. Há especulações, hipóteses, teorias; porém, o fenômeno ainda não foi comprovado sem margens para dúvidas.

Em outras palavras, os eventos $S_1.P_{sm}$ OU $S_2.P_{sm}$, denotado na ilustração $((S_1+S_2).P_{sm})$, são totalmente dependentes da quantidade de neuroides pulsando em S_1 ou em S_2 , em associação com uma certa quantidade de neuroides pulsando em P_{sm} . Porém, tão importante quanto o número de neurônios disparados é o tempo da ocorrência dos eventos. Para que os pulsos coincidam em $((S_1+S_2).P_{sm})$ é necessário que exista uma relação temporal entre as assembleias participantes.

Por exemplo, se P_{sm} acontecer com alguns milissegundos de atraso ou antecipação em relação às outras assembleias, nenhum dos eventos $(S_1.P_{sm}$ OU $S_2.P_{sm})$ poderá ocorrer. Nesse caso, diz-se que P_{sm} sofreu um deslocamento de fase em relação aos eventos internos (S_1+S_2) .

Há pelo menos quatro estratégias para tratar deste problema: a primeira pode ser chamada “*solução espacial*”, que consiste em espalhar conexões sinápticas do evento (S_1+S_2) com diferentes tempos de propagação para muitos neurônios numa vizinhança; assim, a qualquer tempo que o evento P_{sm} ocorrer, sua relação temporal deverá encontrar um ou vários dos neurônios espacialmente distribuídos. O evento como um todo $((S_1+S_2).P_{sm})$ pode acontecer numa vizinhança, em neurônios dentro de uma certa região compatível com o tempo de chegada do evento disparador. Considerando que os próprios neurônios podem fazer adaptações temporais em sua árvore dendrítica, esta solução parece plausível, porque vários neurônios numa vizinhança podem se adaptar a intervalos mais prováveis de deslocamento na fase.

Uma segunda estratégia pode ser chamada de “*solução incidental*”, que consiste em esperar que uma coincidência ocorra. Como o evento P_{sm} tem por natureza ser repetitivo, o sistema pode esperar que uma das ocorrências de P_{sm} coincida com os disparos de S_1 ou de S_2 em alguns poucos neurônios numa região mais restrita. Esta solução estocástica é dependente da repetição, o que significa que o evento de disparo deve ser repetido na esperança que uma de suas ocorrências possa gerar o evento total. Esta estratégia pode falhar ou o evento pode demorar a acontecer. Porém, quando o tempo de resposta não for crítico, ou quando a assembleia representa algo de menor importância, esta é uma solução aceitável e, aparentemente, mais simples. Por outro lado, a estratégia pode também funcionar como filtro, eliminando eventos intermitentes ou ilegítimos: só no caso de P_{sm} ser

mesmo persistente é que ocorrerá sua detecção; dessa forma, ficam fora todos os eventos espúrios indesejáveis.

Uma terceira estratégia pode ser chamada de “*prontas para disparo*”: ela consiste em ter a ramificação biestável S_1 - S_2 disparando tão rápido que a qualquer tempo que P_{sm} ocorrer acontece a coincidência e o disparo de $((S_1+S_2).P_{sm})$. Na verdade, esta é uma solução extrema e provavelmente requer um gasto maior de energia, porque o GPB S_1+S_2 deve pulsar em alta frequência. Esta solução significa que o circuito interno está com *toda a atenção* voltada para qualquer evento que ocorrer em qualquer tempo. Essa solução extrema de operação foi testada nas experiências com assembleias operando como contadores binários [48].

A quarta estratégia pode ser chamada de “*solução sincronizada*”, que consiste em utilizar sinais internos (*pacemaker* ou CPG – como na figura 4.1) que induzam a ocorrência de assembleias em janelas de tempos estritas; o que pode forçar o surgimento de eventos síncronos.

A lógica envolvida é simples: se um sinal S_{yn} interno for adicionado ao circuito detector de cheiro, no exemplo da geração de P_{sm} , o evento $(S_{yn}.P_{sm})$ acontecerá sempre quando o sinal excitatório S_{yn} estiver mais positivo. Isso significa que, havendo um “sincronismo” de P_{sm} com relação a S_{yn} , a relação temporal do evento $(S_{yn}.P_{sm})$ com S_1 e S_2 será favorecida pelo próprio sinal interno S_{yn} , especialmente se ele for aplicado também a S_1+S_2 . Essa solução provavelmente elimina as anteriores dado sua facilidade de implantação e sua robustez; porém as anteriores não podem ser descartadas e provavelmente existam em SNs naturais.

Há várias áreas de pesquisas que consideram sinais internos no controle de redes neurais. Alguns *insights* nesta direção podem vir de linhas de pesquisas como redes oscilantes (*oscillatory networks*) [76], [75]; redes PLL (*phase-locked loop networks*) [15], [16], [17]; osciladores acoplados por pulso (*pulse coupled oscillators*) [72], [73], [74]; sincronização de redes (*network synchronization*) [77]; ou ainda sincronização global em redes com retardo (*global synchronization in delayed neural networks*) [78], [79]. A associação de assembleias biestáveis com geradores de padrões de disparo (neurônios *marca-passo*, CPGs, ou GPBs) promete ser um campo fértil para

novas investigações e para criação de algoritmos complexos usando computação por assembleias.

5.5 Resumo do capítulo

Neste capítulo foi mostrado como assembleias podem computar e executar algoritmos complexos. Assembleias são disparadas por eventos que acontecem na rede. Eventos podem ser neurônios isolados disparando rajadas ou trens de pulsos (*bursts*), ou uma coincidência temporal de certo número de pulsos, ou grupos de neurônios que entram em sincronismo, ou ainda grupos policronizados, entre outros. Quando um número de *spikes* atinge um grupo de neurônios (poli)-sincronamente, acontece uma assembleia ou uma coalizão neural.

Assembleias são fenômenos efêmeros; assim, não faz sentido acontecerem e imediatamente desaparecerem. As coalizões começam a fazer sentido quando interagem com outras assembleias realizando funções com alguma lógica interna. Dessa forma, foi mostrado no capítulo que assembleias podem realizar as funções lógicas necessárias para executar algoritmos digitais (*E*, *OU*, *NÃO*, *NÃO-E*, *NÃO-OU*; chamadas aqui *pseudo-booleanas* em virtude de sua característica estocástica que as difere das reais funções booleanas).

Mais que funções lógicas apenas, coalizões podem realimentar outras coalizões formando laços biestáveis. Dessa forma, as assembleias biestáveis funcionam como “flip-flop” digitais, memorizando um bit de informação (o evento que as disparam). Em outras palavras, assembleias biestáveis (ou GPBs) são os equivalentes aos circuitos sequenciais em eletrônica digital.

Os *eventos* disparadores de assembleias são gerados por padrões internos ou por estímulos externos; assim, os flip-flops neurais *memorizam* um evento que está ligado àquilo que o gerou, ou seja, as assembleias biestáveis *representam* o evento que os provoca. Em outras palavras, GPBs podem representar ações e objetos (fenômenos) externos porque foram justamente estes fenômenos que geraram os eventos disparadores dessas assembleias. Da mesma forma ocorre com os eventos

gerados internamente, ou seja, como estados internos (ex: fome, atenção, sede, etc) ou possivelmente com ideias, abstrações, etc.

Como assembleias são eventos, toda vez que um dos componentes de uma assembleia (biestável ou complexa) disparar ele pode gerar disparo de outros ramos de assembleias. Esse processo de ramificação de assembleias equivale a disparar processos em computação paralela. É o processo de ramificação que permite “desenhar” um algoritmo na medida em que cada novo componente na solução de um problema se torna um ramo formado por uma nova assembléia.

Por outro lado, da mesma forma que podem gerar eventos excitatórios e se ramificar, coalizões podem gerar eventos inibitórios, que silenciam ramificações e outras assembleias e/ou vetam a ocorrência de certos eventos.

A execução de algoritmos complexos (a computação) advém da interação entre assembleias que acontece por meio da associação de todos estes elementos: as funções lógicas operadas no momento do surgimento das coalizões, associadas às reverberações das assembleias biestáveis. A combinação desses elementos tem potencial para realizar, em primeira vista, qualquer tipo de *computação* ou executar qualquer tipo de *algoritmo* e processamento paralelo, embora essa inferência requeira comprovação.

Contudo, computação por meio de assembleias não é simples. Três classes de problemas podem ser destacadas, ao menos até este estágio da investigação.

O primeiro tem a ver com multi-estabilidade das assembleias; ou seja, com coalizões funcionarem em múltiplos estados. Neste caso, assembleias tanto podem reverberar em frequências diferentes, como podem assumir topologias complexas. Em qualquer dos casos, escolheu-se aqui investigar apenas as assembleias biestáveis. Durante o capítulo foram discutidos os diversos problemas de operações multiestáveis, tanto em circuitos com múltiplos atratores operando em frequências múltiplas quanto nas assembleias com topologias e comportamento complexos.

O segundo tipo de problema surge porque eventos disparadores geralmente são repetitivos. A recorrência de eventos pode transformar assembleias biestáveis em osciladores acoplados por pulsos. Mas, as assembleias, como as descrevemos aqui, se comportam de forma passiva, e não são sistemas ou redes osciladoras naturais.

Dessa forma, não parece adequado tratar assembleias como Rede Neural de Pulso Acoplado (RNPA). Assembleias que respondem a pulsos recorrentes têm tendência a entrar em ressonância em altas frequências, similar a sintomas apresentados em redes naturais em crises epiléticas.

O terceiro problema tem a ver com o deslocamento temporal dos eventos disparadores com relação a sinais internos, ou *phase-shift*. A relação temporal é importante para disparar, ramificar ou inibir assembleias; ou seja, para toda a computação e execução de algoritmos por assembleias neurais como a propomos nesta tese. Assim, deslocamentos de fase podem interferir na interação entre GPs, não permitindo que as funções aconteçam ou mesmo provocando comportamento caótico entre as assembleias. Durante o capítulo foram discutidas algumas estratégias para lidar com *phase-shift*, e evitar que provoquem funcionamento inadequado das assembleias.

Dentro do prazo disponível para uma pesquisa de doutorado, até onde seria possível investigar as capacidades, as hipóteses, os casos de plasticidade, e todos os problemas vislumbrados na abordagem da computação por assembleias?

O capítulo seguinte mostra algumas simulações executadas no sentido de validar algumas das proposições e hipóteses dessa tese.

6 SIMULAÇÕES E RESULTADOS

//

Computação por assembleias é um novo campo de investigação com inesgotáveis fontes para novas questões, novos *insights*, novas abordagens e, em especial, para novos experimentos, uma vez que são inúmeras as possibilidades de arranjar assembleias em ramificações que se bifurcam e se inibem, criando algoritmos dos mais simples aos mais complexos.

Dentro da postura de uma abordagem teórica adotada para esta tese, que tipo de simulação seria possível executar que corroborasse as hipóteses aqui apresentadas sobre a computação em assembleias?

Levantamos hipóteses sobre as assembleias executarem algoritmos em virtude de algumas características: (1) no momento em que elas se formam ou são destruídas uma função lógica pseudo-booleana (*AND*, *OR*, *NOT*, *NAND* ou *NOR*) é executada; (2) assembleias podem reverberar e assim funcionam como circuitos recorrentes digitais, como flip-flops; (3) na medida em que assembleias biestáveis se formam passam a operar como memória binária, sem requerem mecanismos de plasticidade; (4) a ramificação das assembleias cria uma relação causal e um controle hierárquico sobre as novas coalizões formadas, essa combinação resulta um algoritmo.

Dessa forma, decidiu-se que o melhor a fazer seria simular e demonstrar que tais propriedades e funções de fato acontecem em redes neurais pulsadas. Afinal, são essas funções e propriedades que embasam a abordagem da computação por assembleias. Infelizmente, não é possível, nem viável, simular todos os mecanismos de plasticidade discutidos anteriormente. Integrá-los nas simulações deve ser visado como objeto de futuros estudos. É importante tê-los apresentado como componentes teóricos que influenciam na computação e que não devem ser desprezados. Mas, o foco desta tese será mesmo as simulações de funções que embasam o corpo teórico da computação por assembleias.

Há na tese cinco apêndices, que são programas Matlab, com os quais se pretende demonstrar e validar as hipóteses levantadas e descritas até o momento.

Será descrito a seguir como o programa do APÊNDICE ‘A’ cria base para operação com estímulos externos à rede pulsada; ou seja, como é possível fazer conversão ou *transdução* de fenômenos físicos do mundo para *spikes*, de forma que a rede possa computá-los. Na sequência, será mostrado como o programa do APÊNDICE ‘B’ foi usado para simular contadores binários e acumuladores; um circuito clássico da eletrônica digital que se mostra complexo para coalizões, uma vez que envolve aspectos como auto-dissociação de assembleias e acoplamento de pulsos com deslocamento de fase.

Posteriormente, o programa do APÊNDICE ‘C’ mostra uma classe de problemas que surgem quando se inserem atrasos em assembleias contadoras. O programa é similar ao do APÊNDICE ‘B’, porém, com retardo entre as assembleias. Por fim, será mostrado que os programas apresentados nos APÊNDICES ‘D’ e ‘E’ são usados para testes dos princípios de operação das coalizões, bem como podem ser uma ferramenta para criar protótipos, e testar soluções antes de modelar algoritmos complexos em assembleias. Primeiramente, vamos descrever aspectos gerais das simulações, usados em todos os programas Matlab apresentados nos anexos.

6.1 Ambiente de simulação

Seja N o número total de neuroides numa rede, que tem o vetor coluna $\mathbf{v}[\mathbf{v}_1, \mathbf{v}_2 \dots \mathbf{v}_N]^T$, $\mathbf{v}[i] \in \mathbb{R}$, representando o potencial de membrana de cada neuroide. A resolução temporal nas simulações (passo no tempo discreto) é $\Delta t = 1$ ms. Nos programas apresentados nos apêndices, o potencial de membrana será calculado de acordo com as equações do “*simple model*” de Izhikevich (ver tópico 2.7.3 da tese):

$$\mathbf{v}' = 0.04 \mathbf{v}^2 + 5\mathbf{v} + 140 - \mathbf{u} + \mathbf{I} \quad (1)$$

$$\mathbf{u}' = \mathbf{a}(\mathbf{b}\mathbf{v} - \mathbf{u}) \quad (2)$$

seguido pela regra de *reset* depois do *spike*: *if* $\mathbf{v} \geq 30$ mV, *then* $\{\mathbf{v} \leftarrow c; \mathbf{u} \leftarrow \mathbf{u} + d\}$

O vetor coluna $\mathbf{I}$ é usado para duas funções: (1) para entrada de correntes sinápticas em um dado neuroide; e (2) para injeção de ruído na rede. Em ambos os casos, o vetor $\mathbf{I}$ representa a corrente em pA que está entrando no neuroide, tanto por

estímulo sináptico quanto por injeção de corrente contínua. Note que na equação diferencial (1), obtém-se como resultado um diferencial de tensão: $v'=(dv/dt)$. Quando este diferencial é multiplicado pelo valor da capacitância da membrana (C), obtém-se a corrente da membrana: $I_c=C.dv/dt$. O valor da capacitância foi transferido para o termo à direita na equação (1) no “*simple model*”, razão pela qual a corrente (vetor I) está sendo somada a termos de tensão ($k_1.v^2$, ou $k_2.v$). Isso pode causar estranheza, como aconteceu a um revisor num dos textos que enviamos para um congresso, mas k_1 e k_2 incorporam os valores da capacitância da membrana, validando a operação na equação.

É possível calcular o potencial da membrana por meio de outro modelo neural que possua mecanismo de geração de *spikes* (ver 2.7 e subtópicos). Os modelos *quadratic I&F* (QIF) e o *adaptive Exponential Integrate-and-Fire* (aEIF) são bons candidatos para análises comparadas (ver tópico 2.7.1).

Para isso, o usuário dos programas dos anexos precisa trocar as equações, adaptar os vetores de entrada I , e reescrever as equações dos mecanismos de recuperação de membrana. Os algoritmos continuam válidos. Isso não será feito nesta tese por falta de espaço; assim, o estudo comparado fica como sugestão para outras investigações, possivelmente, como um objeto de análise para um mestrado.

• **Modelagem topológica:**

Como foi descrito no corpo teórico, dois fatores devem ser levados em conta na modelagem topológica da rede: os *pesos sinápticos* entre neuroides e os *retardos na propagação* dos *spikes*. De agora em diante, devemos considerar o neuroide pré-sináptico com o índice i e o pós-sináptico com o índice j .

• *Do retardo na propagação axônica* $DA_{[i,j]}$:

Considerando a possibilidade de todos os neuroides estarem interconectados e que existe um retardo constante, natural da *propagação axônica* entre neuroides, utiliza-se a matriz $DA_{N \times N}$ (*delay axonal*) para memorizar tais retardos tal que:

$DA_{[i,j]} = 0$ indica que não há atrasos axonais entre i e j . O programa usa $\Delta t=1$ ms (default);
 $DA_{[i,j]} = \delta$; com $\delta \in \mathbb{Z}$, indica que há um atraso de δ ms entre os neuroides i e j .

- *Da modelagem dos pesos sinápticos $Sw_{[i,j]}$:*

Se todos os neuroides podem ser interconectados, considere a matriz $Sw_{N \times N}$ para representar todos os pesos sinápticos; onde $Sw_{[i,j]} \in \mathbb{R} \mid 0 \leq Sw_{[i,j]} \leq sw_{Max}$ denota o peso sináptico do neuroide i (linha na matriz) para o neuroide j (coluna da matriz). O parâmetro sw_{Max} pode ser ajustado no modelo toda vez que forem usadas as regras de plasticidade. Experimentalmente, obtivemos alguns valores para sw_{Max} :

$sw_{Max} = 1$; para simulações com grande número de neuroides por assembleia;

$sw_{Max} \cong 10$; para assembleias com poucos neuroides (de 3-10 neuroides/assembleia);

$sw_{Max} \cong 75$; para sinapses elétricas (*gap-junctions*).

- *Da injeção de ruído na rede $I_{[i,j]}$:*

Uma das formas de verificar se há problemas de estabilidade com as assembleias é por meio de injeção de ruído aleatório em todos os neurônios, representando um sinal natural que se propaga no meio. No Matlab, usamos a função:

```
I=noiseFactor*randn(Nn,1); % entrada de pequenos sinais de ruído (pA)
```

e ajustamos o valor da variável ***noiseFactor*** para testar vários valores de ruído.

Nos programas dos apêndices, os parâmetros importantes para a simulação, como o tempo a ser simulado, o número de neuroides por assembleias, o modo de simulação, o tempo e a frequência dos eventos de estímulo etc., estão todos no começo dos programas. Há um cabeçalho explicando a finalidade do programa e uma breve explicação dos parâmetros e suas influências na simulação. Todos os programas geram gráficos (uma figura) do Matlab ao final. Sobre estas figuras, algumas reproduzidas a seguir, vamos fazer uma análise, geralmente qualitativa, daquilo que foi simulado.

6.2 Transdução de sinais

O mundo exterior aos SNs está cheio de fenômenos físicos que precisam ser convertidos para impulsos elétricos nervosos a fim de serem processados por qualquer aparato cerebral. Como esta pesquisa está voltada para redes neurais

pulsadas, é basal a função de investigar como fenômenos físicos podem ser transformados em *spikes* para os neuroides.

Não vamos adentrar na área da transdução biológica, ou seja, como acontece a conversão de um tipo de energia em impulsos elétricos nos neurônios naturais. Para isso, o leitor pode consultar livros clássicos de introdução à neurociência, já referenciados anteriormente [3], [4], [5], [6]. Três livros de especial interesse sobre o assunto são: “*Biology of Sensory Systems*”, de Smith [132]; “*The Evolution of Organ Systems*”, de Schmidt-Rhaesa [133]; e “*Animal Eyes*”, de Land e Nilsson [134].

Dado um valor representando uma grandeza qualquer, para a tese e para fins em engenharia, interessa analisar como é possível converter essa quantidade em corrente para um neuroide, transdução de energia, e como esse neuroide pode converter esta corrente de entrada em *spikes* para a rede neural. O problema da transdução física de um fenômeno em tensão ou corrente elétrica é bem dominado na engenharia eletrônica; assim, vamos nos concentrar em escolher um tipo de neuroide capaz de transformar uma corrente em disparos. Para esse fim, foi criado o programa do APÊNDICE ‘A’.

Neste programa, escolhem-se os seguintes parâmetros:

```
ttime=500; % Tempo da simulação (ms) para cada I aplicado ao nrd
Imin=.1;   % menor val de corrente Iin a excitar os nrds (pA)
step=.5;   % incremento na corrente In a cada ttime
Imax=25;   % maior val de corrente Iin a excitar os nrds
simMod=0;  % modo de simulação: = 0 sem ruído; In é CC In(DC)
           = 1 In(DC)+ruído (% do sinal In DC)
           = 2 In*(DC)+ruído (.05 pA randomico dist. normal).
```

ip=1 % ATENÇÃO: ESCOLHER um indx para o tipo de NEUROIDE a ser testado

O usuário escolhe um índice (variável *ip*) que aponta para um conjunto de parâmetros na tabela *Pars*[], que contém os valores para *a*, *b*, *c*, *d* e *v* inicial da eq.(1) e eq.(2) do *simple model*. A matriz de parâmetros *Pars*[] representa um dos 20 tipos de neuroides apresentado na figura 2.1, mais três tipos especiais de neuroides. Fazendo *ip*=7 será simulado o neuroide tipo “G” da figura 2.1, conhecido como “*Class 1 excitable*”; enquanto *ip*=8 simula o neuroide tipo “H”, ou “*Class 2 excitable*”.

O usuário tem a possibilidade de escolher então o valor mínimo de corrente (*Imin*) que será aplicada ao neuroide, assim como o valor máximo de corrente (*Imax*) e um valor para o degrau (*step*). O programa começa injetando na entrada do neuroide

escolhido o valor de corrente ***I_{min}***, com ou sem ruído – veja ***simMod***, incrementa a corrente injetada do valor ***step*** e repete até o valor máximo ***I_{max}***. Depois o programa mostra o gráfico com as respostas neurais (média de disparos) ao final.

Há dois intervalos de excitação interessantes para análise, conforme descrito no cabeçalho do programa:

- (1) de 0.1 pA até 25 pA - step = 0.1 (para análise de acoplamentos fracos)
- (2) de 1 pA até 100 pA - step = 0.5 (para análise de acoplamentos fortes)

Os gráficos foram obtidos a partir da injeção de um nível de corrente no neuroide, e durante 500 ms (variável ***ttime***) foi calculada a média de disparos do neuroide.

Figura 6.1 **Curvas *IxF***. Alguns neuroides apresentam curvas de resposta aproximadamente lineares em função da corrente de excitação que recebem. Neuroides do tipo 07 e 12 começam a responder acima de 25 pA de excitação, suas curvas são crescentes e aproximadamente lineares. Já o neuroide do tipo 08 tem resposta aproximadamente linear de 0,3 pA até 24 pA. O neuroide do tipo 17 é pouco indicado para transdutor, pois sua curva de respostas ***IxF*** não é monotônica estritamente crescente.

O programa incrementa ***I*** e calcula a nova média em ***ttime*** ms; e assim o faz durante o intervalo de ***I_{min}*** até ***I_{max}***. A figura 6.1 mostra as curvas dos neuroides (***ip***=7, 8,

12), bons conversores $I \times F$ (de corrente I para frequência média F) numa faixa linear de disparos; curvas que podem ser confrontadas com a do neuroide $ip=17$.

Note que os neuroides do tipo 7 e 12 apresentam respostas parecidas, com curvas aproximadamente lineares, monotônicas e estritamente crescentes na faixa de excitação de 25 até 100 pA. Por outro lado, o neuroide do tipo 8 responde também de forma aproximadamente linear e monotônica na faixa de excitação de 0,3 pA até $\cong 24$ pA; acima desse valor ele não responde linearmente. O neuroide 17 não tem curva linear monotônica, o que sugere que pode ser complicado usá-lo como transdutor de entrada de sinais.

Figura 6.2 **Curvas $I \times F$ em baixo nível de excitação.** Para correntes entre 0,1 e 25 pA, alguns neuroides (7, 12, 23) permanecem silentes até perto de 25 pA. Vários neuroides (4, 6, 10, 14) respondem de forma instável. Alguns têm respostas na forma de escadas (6) ou patamares (4). Outros (11, 21, 22) respondem linearmente dentro de certa faixa. Boa parte responde com certo grau de linearidade, apresentando curvas semelhantes às mostradas na figura para os neuroides 1, 9 e 15.

Vários neuroides (1, 2, 5, 9, 13 e 15) apresentam curva aproximadamente linear quando excitados por baixos níveis de correntes (0.1 até 20 pA). Outros tipos de

neuroides (11, 21, 22) apresentam faixas de respostas lineares para estes níveis de corrente. Algumas curvas são apresentadas na figura 6.2. Veja que os neuroides 4, 6 e 7 possuem curvas inapropriadas para conversores nesta faixa de excitação.

Alguns neuroides apresentam curvas IxF peculiares, que requerem alguma atenção. Em primeira análise, estas curvas depõem contra o uso da média de disparo para computação em redes neurais. Razões que comprometem a adoção de taxa de disparo já foram apontadas no tópico 2.4; e.g., não há tempo suficiente para calcular média em certas operações computacionais realizadas pelo cérebro. Somado a isso, é possível inferir que alguns saltos mostrados na curva IxF não são compatíveis com a “bem comportada” curva sigmoidal usada como função de ativação em RNA não pulsadas. A figura 6.3 ilustra alguns comportamentos nos quais a média de disparos revela saltos associados a regiões ou patamares estáveis.

Figura 6.3 **Curvas IxF não lineares.** Os neuroides da figura apresentam faixas nas quais variações na corrente não representam modificação na frequência; ou seja, apresentam patamares de operação que independem da corrente de entrada (veja neuroide 4 sobreposto ao 10, e neuroide 14). Esses neuroides são candidatos a atuarem como “chaves” biestáveis, indicando se um evento externo está presente, respondendo de forma estável sempre na mesma frequência. O neuroide 8 é linear para baixas correntes, pode atuar como chave entre 50-95 pA (resposta de $\cong 100$ Hz), e acima desse valor de corrente ele dispara em $\cong 500$ Hz. O neuroide 22 tem a forma da curva parecida com o neuroide 8, com patamares e faixas de corrente diferentes.

A interpretação para os saltos é que alguns neurônios (3, 4, 14) respondem emitindo rajadas (*burst*). A rajada pode ter um papel importante, como o de garantir que certo

evento importante na rede receba atenção. Se ocorrer uma rajada e, na sequência, o neuroide silenciar por um período maior que o da rajada, como é o caso dos neuroides 4 e 14, sua média será abaixo de 50%. Explica-se com isso que, o comportamento de alguns neuroides é qualitativamente mais importante para a rede do que quantitativamente. Em outras palavras, uma rajada pode ser decisiva, ela pode funcionar como “*interrupt request*” nos computadores digitais, uma requisição de atenção imediata. Já sua média pode apresentar valores estranhos que dependem do período no qual a rajada aconteceu dentro do cômputo da média.

Em resumo, o programa do APÊNDICE ‘A’ permite escolher neuroides para entradas das redes pulsadas, e ao mesmo tempo analisar faixas operacionais de alguns tipos de neuroides individuais, de forma a prever seu comportamento na rede. Para uma junta robótica, por exemplo, podemos usar um neuroide com curva linear, como os do tipo 7 e 12 na figura 6.1, excitando-os com correntes entre 30 e 90 pA, obtendo com isso respostas proporcionais entre 20 e 120 *spikes*/segundo. Por outro lado, abaixo desse valor estes neuroides ficariam silentes. Os neuroides 8 e 22 seriam indicados para transdução em faixas de correntes baixas. Os neuroides 8, 10 e 22 podem ser úteis como chaves digitais ON/OFF; uma vez que, excitados entre 30 e 70 pA, respondem numa faixa constante de $\cong 150\text{Hz}$. Requisição de *atenção* da rede por meio de rajadas, uma espécie de “pedido de interrupção”, pode ser executada pelos neuroides do tipo 4 (ativado por excitação) e do tipo 14 (ativado por inibição); embora os anteriores (tipos 8, 10, 14) também possam realizar função semelhante de forma persistente, em vez de rajadas.

É muito rico o repertório com os 23 tipos de neuroides simulados pelo *simple model* e um engenheiro eletrônico terá em mãos vários componentes, com respostas de natureza diferentes para interligá-los em um sem número de combinações. Este programa permite ter uma prévia da resposta do neuroide. Um programa mais elaborado, embora não calcule curvas de respostas $I \times F$, mas mostre as resposta individuais dos neuroides do *simple model* pode ser obtido em:

<http://www.izhikevich.org/publications/whichmod.htm> (acessado em 15/10/2011)

Faça download do arquivo izhikevich.m. O programa possui interface amigável na qual se vê o comportamento dos neuroides do *simple model* operando como sistema dinâmico, com janelas de *phase portrait*, gráficos para V e U , etc. (ver [26]).

6.3 Contadores digitais binários e acumuladores

Um caminho natural quando se trabalha com sistemas complexos é manter constante alguma variável ou parâmetro enquanto outros são avaliados, mantendo o domínio da experiência restrito. Um dos problemas que inicialmente foram pensados como importantes para assembleias era o ruído natural das redes populosas, que são frequentemente relatados em neurociências. Por outro lado, nossa hipótese é que assembleias executam funções lógicas aproximadamente iguais às booleanas, e que também memorizam, operando como flip-flops. Dessa forma, a fim de realizar uma experiência que mostrasse a potencialidade das assembleias, simulamos contadores binários digitais, num primeiro momento sem considerar atrasos na propagação. Além disso, as assembleias foram simuladas sincronamente para facilitar visualização; ou seja, a simulação foi feita com *synfire chains* sem atrasos.

A razão pela escolha dos contadores binários é simples: eles requerem portas lógicas que executem funções *OU*, *E*, e *NÃO*; precisam executar auto-inibição, além de memorizar informação. Esta experiência está relatada em detalhes em [48]; portanto, vamos discutir apenas aspectos destacáveis da experiência, cujo programa se encontra no APÊNDICE 'B', e pode ser testado pelo leitor.

O programa possui alguns parâmetros que o usuário poderá alterar:

```
simMode=1;           % simulation mode: 1=BinaryCounter  0=Stacker
GlobalResetTime=0;   % Global Reset time burst (=0 not simulated)
Nk=5;                % Num of neurons per kernel (3...12 neurons)
pti=80;              % Pulse Time Interval (ms) (for Input pulse)
NoiseFactor=0.05;    % Fator de ruído introduzido ao circuito...
simTime=1600;        % tempo total de simulação
```

A variável ***simMode*** muda o modo de simulação entre contador binário (=1) e acumulador ou *pilha* (=0). Quando estiver usando acumulador, um *reset* global pode ser aplicado para desarmar as assembleias. Para isso, ajuste o tempo da variável ***GlobalResetTime*** para um valor em *ms* no qual se deseja aplicar o *reset*. A variável ***Nk*** controla o número de neuroides da assembleia (chamada *kernel* no texto original

[48]). A variável *pti* controla o intervalo de tempo entre os estímulos a serem contados pela rede (veja acima que os pulsos se repetem a cada 80 ms). A variável *simTime* controla o tempo total (em ms) que o programa simulará.

Como foi dito, um dos objetivos iniciais foi estudar até que ponto a assembleia se mantém ativa em função dos níveis de ruídos injetados na rede; assim, um parâmetro “fator de ruído” pode ser modificado por meio da variável *NoiseFactor*.

A topologia da rede é composta de quatro *pools* fixos de neuroides, sendo que cada *pool* realiza a função de um flip-flop J-K operando de forma complementar²⁸. Cada *pool* é constituído de quatro *kernels* denominados *IN*, *K₁*, *K₂* e *K₃*. Eles são interconectados como mostrado na figura 6.4.

Figura 6.4 **Topologia do contador binário.** A figura mostra 2 dos 4 *pools* usados para contador binário. Cada *pool* tem quatro *kernels* (*IN*, *1*, *2*, e *3*), e cada *kernel* contém um número (*Nk*) ajustável de neuroides totalmente conectados entre si. A rede conta o número de eventos de estímulos ocorridos. Um *reset* global desarma todas as assembleias, independente do estágio dos contadores.

O *kernel IN* é uma assembleia de entrada, um grupo de inter-neuroides que recebe estímulo externo e dispara internamente *K₁*. Os *kernels K₁* e *K₂* formam uma

²⁸ Flip-flop (FF) J-K operando em forma complementar significa que as entradas *J* e *K* desse circuito eletrônico foram colocadas em nível lógico alto (=1). Quando um pulso de *clock* chega ao FF, a saída *Q* assume o valor complementar $\bar{Q}$, enquanto a saída $\bar{Q}$ assume o valor de *Q*. Quando outro pulso de *clock* atinge o FF ocorre o mesmo processo; assim, tanto $\bar{Q}$ quanto *Q* voltam aos seus valores originais. O FF se mantém alternando entre valores complementares a cada pulso de *clock*.

assembleia biestável. Todos os neuroides de K_1 estão conectados a K_2 que, da mesma forma, tem seus neuroides conectados de volta a K_1 , formando um *loop*.

Tanto K_1 quanto K_2 alimentam K_3 , mas, isoladamente, não conseguem fazer esta assembleia disparar. O *kernel* K_3 é um grupo de *reset* interno, ou seja, faz a auto-inibição da assembleia biestável K_1 - K_2 . Para que K_3 dispare é necessário receber pulsos das assembleias K_1 OU K_2 , associado (coincidente) com um estímulo na assembleia de entrada *IN*. Ou seja, o *reset* interno (que servirá também para estímulo do próximo *pool*) acontece quando:

$$\overline{K_3} = K_1 \cdot IN + K_2 \cdot IN$$

Note que a função do *kernel* K_3 é inibitória internamente. No modo de simulação *simMode*=0 (acumulador ou pilha) o *kernel* K_3 perde a função inibitória interna, a topologia muda para uma configuração mais simples, na qual os *kernels* K_1 e K_2 formam uma assembleia que retém indeterminadamente um impulso excitatório.

Figura 6.5 **Assembleia como acumulador**. Esta é uma figura do Matlab modificada de forma que os *kernels* foram destacados em cores diferentes. Os *kernels* $N-1$ ($N = pool$ ao qual o *kernel* pertence) estão na cor azul enquanto os *kernels* $N-2$ estão na cor vermelha. Na figura, cada assembleia possui 5 neuroides e todos foram resetados aproximadamente em 550 ms (*GlobalResetTime*=550).

A figura 6.5 mostra a resposta dos disparos para o modo de operação *simMode*=0 (acumulador). A figura, um *raster plot* dos neuroides, foi modificada e colorida a fim de destacar os *kernels*, facilitando a visualização. Os *kernels* K_1 e K_2 do *pool 1* (o *pool* de entrada) estão denominados **1-1** e **1-2**, respectivamente. Assim, os neuroides na base da figura são os de entrada *IN* do *pool 1*, denominados **1-IN**.

Os neuroides **1-IN** ativam **1-1**, que estão em azul. Os neuroides **1-1** excitam **1-2**, que estão em vermelho. Dessa forma **1-1** e **1-2** formam uma assembleia biestável. Nesse tipo de contador, depois de disparadas as coalizões permanecem ativas até serem desarmadas pelo *Global Reset Event*.

A condição para que o *pool 2* se torne ativo é que os neuroides do *pool* anterior (**1-1** e **1-2**) estejam ativos **E** que um novo evento de estímulo E_T aconteça na rede:

$$K_{IN}^Q = K_1^P \cdot E_T + K_2^P \cdot E_T$$

onde E_T denota o evento de estímulo (*event trigger*); K_{IN}^Q se refere ao kernel *IN* do *pool* subsequente, enquanto K_N^P denota os *kernels* K_1 e K_2 do *pool* antecessor. Note que após um *reset* global todas as coalizões são inibidas e voltam a acumular, hierarquicamente, ativando as cadeias em função do estímulo de entrada.

A figura 6.6 mostra a resposta dos disparos para o modo de operação *simMode*=1 (contador binário). A figura do Matlab, um *raster plot* dos neuroides, também foi modificada e colorida a fim de destacar os *kernels*, facilitando a visualização.

Um contador binário digital de quatro bits conta do valor 0000_b até 1111_b. Similar ao gráfico anterior, o *pool 1* é representado pelos neuroides inferiores, significando o dígito 2^0 (unidade). O *pool 2* representa 2^1 (peso 2) e o *pool 3* representa o dígito binário 2^2 (peso 4). Por último, o *pool 4* representa 2^3 (peso 8). A lógica de operação é simples, a cada pulso (estímulo) um *pool* deve tornar-se ativo (=1) e memorizar este estado.

Ao receber o estímulo seguinte, ele deve desativar-se, retornando para zero, e permanecer neste estado. Esta operação é semelhante à de um circuito FF J-K complementar. O estímulo é passado para o *pool* seguinte seguindo a regra:

$$K_{IN}^Q = K_1^P \cdot K_{IN}^P + K_2^P \cdot K_{IN}^P$$

onde K_{IN}^Q denota o *kernel IN* do *pool* subsequente, enquanto K_N^P denota os *kernels* K_1 e K_2 do *pool* antecessor, e K_{IN}^P denota o *kernel* de entrada no *pool* antecessor.

No programa do APÊNDICE 'B' o usuário poderá estender o tempo de simulação, *simTime*=2000, por exemplo, para ver o efeito dos quatro *pools* voltando ao 0000_b (todas as coalizões silentes) assim que os *pools* atingirem 1111_b e um novo estímulo for ativado. Na janela de comando do Matlab surge: **Max=0.22 / Min=-0.21** (em pico-Amperes); que se refere aos picos de corrente injetados como ruído no circuito .

Figura 6.6 **Contador binário**. Cada *kernel* tem 3 neuróides. O *pool 1* recebe estímulo externo pelo grupo de neuróides 1-IN. A cada pulso na entrada o *pool 1* se torna ON e no pulso seguinte se torna OFF. Os grupos subsequentes também se comportam dessa maneira, a diferença é que o estímulo dos grupos sucessores vem do *pool* anterior. Essa sequência forma um contador binário (ver texto).

Com 3 neuróides por assembleia, podemos alterar o nível de ruído para um fator até **NoiseFactor**=0,17 (que representa correntes de Max= Max= 0.73 / Min= -0.72 pA) sem que a rede perca a estabilidade; ou seja, a rede permanece contando binariamente até este valor. O valor de intervalo inter-pulsos externos foi de 120 ms.

Com 12 neuróides por assembleia, mesmo intervalo inter-pulsos, é possível alterar o nível de ruído para um fator ainda maior, **NoiseFactor**=0,46 (que representa valores de Max= 2.12 / Min= -2.2 pA) sem perder a estabilidade da rede como contador. Em ambos os casos foram usados *simTime*=2.000 ms.

Ao menos mediante uma análise qualitativa dos dados acima, podemos observar que assembleias são estáveis e que, ao incrementar o número de neuroides por assembleias, a rede se torna mais robusta com relação a ruídos, como se poderia esperar. Antes de passar a uma análise quantitativa e estatística deste circuito, pareceu ser importante inserir na rede o outro componente: os atrasos na propagação dos *spikes*. Ao realizar isso, nos deparamos com outra classe de problemas mais importantes que a análise da robustez da rede modelada de forma incompleta.

6.4 Contadores com retardo e rede de pulso acoplado

O programa apresentado no APÊNDICE 'C' é similar ao contador binário anterior, sendo apenas uma versão modificada na qual foi introduzida a matriz de *delay* axonal **DA**[] que contém o atraso entre todas as conexões. Da mesma forma que o anterior, estamos usando *synfire chain*; assim, a matriz de atrasos foi programada para causar retardo único, cujo valor é ajustável por meio da variável **dIK**. O conjunto de parâmetros do programa é:

```
% -- 1. CONFIGURAÇÃO da simulação -- %
simMod=1;           % modo de simulação: 1=BinaryCounter  0=StackerCounter
dIK=10;             % atraso inter-kernel (inter-neuroids) delay
nK=5;               % Num de nrds por kernel (3 <= nK <= 20 nrds)
fatorRuido=0.05;    % Fator de ruído (random normalz) injetado na rede

% -- parametros temporais da simulação -- %
tsim=500;           % tempo (ms) total de simulação
maxPD=20;           % Tempo máx de atraso na propagação axonal dos spikes
globalResetTime=00; % Tempo (ms) para ocorrer reset global (=0 não simulado)
firstIN=50;         % atraso para o 1o. INPUT (1o. pulso)
pINti=104;          % valor inicial p/ intervalo evento INPUT (será
incrementado)
pINtf=140;          % valor final p/ intervalo evento INPUT
```

Diferente do anterior, este programa foi modificado para demonstrar como atrasos na propagação introduzem respostas diferenciadas nas redes neurais pulsadas, demonstrando a importância da temporalização em RNPs, especialmente na computação por assembleias, nas quais a temporalidade não pode e não deve ser negligenciada.

Se o usuário executar o programa da forma como está listado no anexo obterá a figura 6.7A e uma mensagem na janela de comando do Matlab: "Int 104 - ok".

Como nenhum dos neuroides entre os números 15 e 90 dispara, as imagens foram modificadas para facilitar a visualização e por economia de espaço. Este programa está em *loop*, e o usuário deve clicar sobre a janela da figura para obter o próximo gráfico, que será gerado depois que o programa incrementa o intervalo entre eventos de entrada. Se o usuário quiser interromper a execução do *loop* deve clicar na janela de comando e acionar no teclado *Ctrl + Break/Pause*.

Figura 6.7 **Contadores com retardo entre assembleias.** (A) Este é um *raster plot* do contador, circuito igual ao contador binário, aos quais foram adicionados os atrasos de propagação dos *spikes*. (B) Quando o estímulo de entrada tem uma frequência diferente de um dos harmônicos da reverberação entre assembleias biestáveis, pulsos adicionais são introduzidos entre os *kernels*, o que pode levar a rede a operar de forma imprevisível, numa frequência mais alta, ou ter comportamento caótico. No destaque são mostrados pulsos adicionais introduzidos em $1-K_1$ e repassados para o *kernel* $1-K_2$. A cada novo estímulo de entrada um novo pulso surge entre as coalizões $1-K_1$ e $1-K_2$. (C) A coincidência de pulsos dispara os neuroides do *pool* hierarquicamente superior.

Clicando sobre a figura uma vez faz surgir o gráfico com intervalo de 105 ms e, clicando novamente, surgirá a tela mostrada em 6.7B com a mensagem na janela de comando: "Int 106-ok". Na imagem foi destacado que um deslocamento de 2 ms no intervalo entre eventos de entrada faz surgir um pulso adicional no *kernel* $1-K_1$, que é propagado para $1-K_2$. Esse pulso se agrega ao padrão de disparo do grupo biestável

que passa a pulsar em dobro. Com a chegada de outro evento de entrada um novo pulso é incorporado ao *loop* formado por $1-K_1$ e por $1-K_2$. Assim, clicando na figura algumas vezes o usuário poderá perceber que a assembleia biestável memoriza novos pulsos de entrada, tendendo a atuar em frequência cada vez mais elevada, ou a se desestabilizar.

Repetindo o click sobre a figura algumas vezes, o usuário verá a figura 6.7C com a mensagem “Int 117-ok”. Nesse momento, veja que a coincidência de pulsos de entrada com estados internos da assembleia $1-K_1$ e $1-K_2$ fazem a assembleia $2-IN$ disparar, acionando assembleias do *pool* superior. Mas isso ocorre por acaso, numa situação de coincidência, não que as propagações estivessem programadas para isso.

Com uma pequena modificação no programa, impedindo eventos de estímulos de entrada após certo tempo, foi obtida a figura 6.8 que destaca como a assembleia biestável “memoriza” novos interpulsos em sua reverberação a qual permanece constante num novo padrão de disparo.

Figura 6.8 **Pulso acoplado às assembleias biestáveis.** As assembleias tendem a memorizar novos eventos de entrada incorporando-os em sua reverberação. A tendência é a reverberação numa frequência que aumenta a cada novo pulso introduzido. O experimento mostra a capacidade dos neuroides de participarem de outras assembleias, compartilhando seu tempo de disparo, apesar de que, neste experimento, eles estejam participando sempre da mesma coalizão.

Esse fenômeno permite demonstrar duas coisas: (1) se uma coalizão biestável for excitada múltiplas vezes, ela tende a agregar novos intervalos interpulsos em seu padrão de disparo. A fim de não entrar em colapso e operar em frequências extremas, deve haver formas das assembleias biestáveis ativas bloquearem novos impulsos, assunto que foi discutido no tópico 5.4.2. É possível tratar esta situação

como Rede Neural de Pulso Acoplado (RNPA) ou investigando outras formas de bloqueio seletivo dos estímulos.

É possível também (2) demonstrar que um grupo de neuroides pode participar de mais de uma assembleia, compartilhando seu tempo de disparos entre uma ou outra coalizão. Claro, no exemplo acima os grupos estão participando da mesma assembleia com múltipla excitação; contudo, é possível inferir que eles poderiam ter sofrido estímulo de outra fonte e terem participado de outras coalizões durante o tempo em que estão silentes, esperando o retorno dos pulsos reverberantes do grupo antagônico dentro da assembleia biestável. Ou seja, da mesma forma que um neuroide pode disparar respondendo dupla ou triplamente a um estímulo de um pulso acoplado numa única assembleia, como acontece no exemplo da figura 6.8, eles poderiam responder a estímulos de outros pulsos acoplados, participando de outras coalizões.

Durante a tese tornou-se nítida a urgência de criar uma ferramenta que permita toda sorte de investigações basais sobre assembleias. Uma ferramenta que facilite, por exemplo, estudar casos isolados das funções pseudo-booleanas, da participação de neuroides em múltiplas assembleias, da formação de grupos policronizados; enfim, os problemas básicos levantados no percurso da abordagem sobre computação por assembleias, para então propor redes mais complexas realizando operações computacionais mais elaboradas. O resultado foi o programa que será discutido no tópico seguinte.

6.5 Simulação de base para assembleias

Na tentativa de solucionar o problema quebrando-o em partes, foi criado um código em Matlab que permite controlar e fazer os testes básicos envolvendo os dois elementos essenciais para computação por assembleias: os *pesos sinápticos* entre neuroides e o *atraso na propagação* de *spikes* entre elas. Este programa está listado no APÊNDICE 'D', com ligeira modificação no APÊNDICE 'E' para operar com grupos policronizados. Nele há alguns parâmetros adicionais a serem configurados e a topologia da rede é desenhada pelo usuário.

Alguns parâmetros, como *tsim*, *simMod*, *tINev*, *tiINev*, *tGReset*, são auto-explicativos e podem ser alterados pelo usuário trocando o valor dessas variáveis.

```
tsim=500; % Tempo da simulação (ms) para cada I aplicado ao nrd
maxPD=22; % max atraso de propagação (delay propagation) simulado
tBD=15; % tempo base para delay entre nrds (i) -> (j) (ms)
tbN=21; % TIPO de nrd (indx p/ Pars) (se preciso, mudar em
Topol]
simMod=1; % modo: =0 sem atraso randômico (synfire chain)
% =1 com atraso rand. (GPs policronizados)

% -- define num assembleias e nrds -- %
nNa=5; % Núm de neuroides/assembleia: 3<=nNa<=20
if nNa<3; nNa=3; elseif nNa>20; nNa=20; end; % restringe nNa
numAssb=5; % TOTAL de Assbs (cuidado p/ não usar indx > em Topol)
nN=nNa*numAssb;
sw0=68/nNa; % peso sináptico inicial p/ todas as conexões

% -- define tempos de estímulos (INPUT)-- %
tINev=40; % tempo no qual ocorre o primeiro evento de INPUT
tiINev=340; % intervalo entre eventos de INPUT (=0 não repete)
tGReset=252; % tempo para o reset global...
f_tINevRnd=0; % flag =1 adiciona aleatoriedade ao evento INPUT
TIEv=tINev*ones(nNa,1); % vetor c/ tempos de disparos dos nrds INPUT
if simMod==1 TIEv=4*randn(nNa,1)+TIEv(:,1); end
TIEv=uint32(TIEv);

% -- desenha TOPOLOGIA da rede %
% ex: '0'->1; '1'->'2'; '2'->'1' (loop); '1'->'3'; '2'->'3'; '4'->'1'; '4'->'2'.
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;...
1 3 sw0/2 1.5*tBD tbN; 2 3 sw0/2 0.5*tBD 0;...
4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0]; % estes últimos são global reset
```

Na definição das assembleias o usuário deve informar quantas assembleias serão usadas na rede, alterando a variável **numAssb=5**. Cuidado para não indexar na topologia um número de assembleia maior que o máximo assinalado em **numAssb**, lembrando que as assembleias serão indexadas a partir do zero. A assembleia zero geralmente é usada para injetar estímulo externo.

Cada assembleia terá **nNa** neuroides, com $3 \leq nNa \leq 200$, mas o usuário pode alterar o limite para mais que 200 se necessário. O peso sináptico default **sw0** é calculado em função do número de neuroides por assembleia. Mas, o peso sináptico individual entre assembleias pode ser alterado pelo usuário ao desenhar a topologia.

Para calcular o atraso dos pulsos entre os neuroides *i* e *j* é necessário buscar no histórico de disparos quais neuroides dispararam algum tempo atrás. Quanto atrás? Para controlar essa busca foi introduzida a variável **maxPD**. Isso significa que o programa fará uma busca nos disparos ocorridos nos últimos **maxPD** ms para ver quais neuroides dispararam. Um valor **maxPD=22** é suficiente para trabalhar com um tempo básico entre atrasos de assembleias de até 20 ms.

A variável **maxPD** pode ser alterada para mais se o usuário quiser trabalhar com atrasos maiores entre assembleias; porém, isso aumenta o custo computacional da simulação. O ideal é que ela fique 2 ms maior que o máximo de atraso entre as assembleias.

O valor padrão de atraso axonal entre assembleias pode ser controlado pela variável **tBD**. No caso de simular grupos de *synfire chain*, por exemplo, com atraso de 12 ms, faça a variável **tBD=12**. O leitor vai se lembrar que em mamíferos pode ocorrer atrasos interneurais de até 44 ms (ver tópicos 2.4 e 3.7.1).

A parte que requer mais cuidado por parte do usuário será na criação da estrutura das assembleias. A topologia da rede é feita por meio da conexão de assembleias, descritas na matriz **Topol []** com o seguinte formato:

$$\text{Topol} = [i_1 \ j_1 \ s_1 \ d_1 \ ip_1; i_2 \ j_2 \ s_2 \ d_2 \ ip_2; \dots; i_n \ j_n \ s_n \ d_n \ ip_n];$$

onde:

i_x = índice da assembleia pré-sináptica,
 j_x = índice da assembleia pós-sináptica,
 s_x = peso sináptico entre os neuroides de $i_x \rightarrow j_x$
 d_x = atraso axonal dos neuroides $i_x \rightarrow j_x$
 ip_x = tipo de neuroide da assembleia j_x (é um índice para a tabela: **Pars[]**)

Cada valor na matriz é separado pelo caractere “*espaço*”, seguido do caractere “;” ao final de cada vetor linha, como é padrão do Matlab.

Todas as conexões entre assembleias devem ser informadas. Dessa forma, para o exemplo da topologia desenhada para o programa do APÊNDICE ‘D’, temos:

```
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;
1 3 sw0/2 1.5*tBD tbN; 2 3 sw0/2 0.5*tBD 0;
4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0];
```

O primeiro vetor linha dessa matriz (0 1 sw0 tBD tbN;) descreve que a assembleia ‘0’ (neuroides 1,2,...,nNa) está conectada à assembleia ‘1’ por meio do peso sináptico descrito na variável **sw0**, com um atraso da assembleia ‘0’ para a ‘1’ de **tBD** milissegundos, e que o tipo de neuroide da assembleia ‘1’ tem o índice **tbN**.

Para manter compatibilidade com a nomenclatura usada nas assembleias dos programas anteriores, vamos chamar a assembleia ‘1’ de **K₁**, assim como as outras coalizões denotadas na matriz; ou seja, **K_n** representa o *kernel n* na matriz de

topologia *Topol*]. Assim, a conexão do primeiro vetor denotado na matriz é K_0 , evento de entrada, que conectado à assembleia, ao kernel K_1 .

O segundo vetor linha na matriz descreve que a assembleia K_1 está conectada com K_2 , depois temos K_2 conectada com K_1 fechando um *loop* biestável. Na sequência, temos K_1 ligada a K_3 , e K_2 também ligada à assembleia K_3 . Ambas as conexões foram feitas com metade do peso sináptico padrão ($sw_0/2$) e ainda os tempos entre $K_1 \rightarrow K_3$ e $K_2 \rightarrow K_3$ foram modificados do atraso padrão. Enquanto o atraso de $K_1 \rightarrow K_3$ é 1,5 vezes o padrão, o de $K_2 \rightarrow K_3$ é só metade do padrão.

Figura 6.9 **Simulador de coalizões APÊNDICE D.** No modo de simulação *synfire*, com 8 neuróides por assembleia, K_0 dispara K_1 que dispara K_2 , que está conectada com K_1 formando um *loop*, uma assembleia biestável. O evento K_3 é resultado de uma função pseudo-booleana *AND* entre K_1 e K_2 ($K_3=K_1.K_2$). Um *reset* global é aplicado pela assembleia K_4 (inibitória) no tempo $tGReset=252$ ms.

Por último, nos dois últimos vetores linha, descreve-se que a assembleia K_4 está conectada com peso sináptico negativo ($-sw_0$), de forma que esta assembleia é inibitória. Os neuróides da assembleia K_4 são os de maior índice na rede, ou seja, estão no topo da rede. A assembleia K_4 inibe tanto K_1 ($K_4 \rightarrow K_1$) quanto a coalizão K_2 ($K_4 \rightarrow K_2$), promovendo desarme (ou *reset*) da assembleia biestável que é formada por K_1-K_2 . Fazendo a variável *simMod*=0 e executando o programa, o usuário obterá um gráfico como o da figura 6.9.

Nesta simulação o programa dispara a assembleia K_0 (com 8 neuroides, de 1...8) que é o evento de estímulo (*input*) da rede. Geralmente, reservamos os neuroides de menor índice na rede para entrada. Este evento de estímulo só volta a disparar perto de 400 ms, uma vez que $tiNev=340$.

Podemos observar como o evento de estímulo aplicado a K_1 faz esta assembleia disparar K_2 que reverbera e dispara K_1 novamente. Sem receberem novos estímulos, as duas formam uma assembleia biestável reverberando e com frequência constante. Note que K_3 só dispara depois da coincidência temporal entre os disparos de K_1 e K_2 , que ocorre $1,5 \cdot tBD$ milissegundos depois do disparo de K_1 , e $0,5 \cdot tBD$ ms depois do disparo de K_2 .

Figura 6.10 **Funções pseudo-booleana OU / E.** (A) Modificando a topologia, diminuindo o peso sináptico entre as assembleias $K_2 \rightarrow K_3$, o evento K_3 deixa de acontecer. Isso comprova que a função *AND* estava sendo executada no gráfico anterior (figura 6.9). (B) Por outro lado, modificando o peso sináptico para o valor default, qualquer das duas assembleias K_1 ou K_2 é capaz, isoladamente, de disparar K_3 ; ou seja, K_1 OU K_2 pode disparar K_3 . No segundo gráfico os tempos entre $K_1 \rightarrow K_3$ e $K_2 \rightarrow K_3$ foram modificados ligeiramente para facilitar visualização.

Como experiência, o usuário pode modificar o peso sináptico entre K_1 ou K_2 com a assembleia K_3 , digamos, baixando o peso de K_2 para $sw_0/5$. O usuário notará que K_3 deixará de acontecer, como na figura 6.10A. A matriz da topologia fica assim:

```
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;
        1 3 sw0/2 1.5*tBD tbN; 2 3 sw0/5 0.5*tBD 0;
        4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0];
```

O fato de K_3 deixar de acontecer significa que não houve excitação suficiente para disparar os neuroides de K_3 . A temporalidade foi mantida, porém os pesos sinápticos mais fracos entre K_2 e K_3 não provocam os disparos.

Isso comprova que é necessário que K_1 E K_2 aconteçam simultaneamente, com os pesos sinápticos $\cong sw_0/2$ em ambas para que o evento K_3 ocorra.

Por outro lado, se o usuário modificar a matriz de sorte que o peso sináptico *default* seja usado, ou seja, de forma que sw_0 não seja dividido em nenhuma das conexões entre K_1 e K_3 e K_2 e K_3 , qualquer dos grupos terá, isoladamente, força para disparar K_3 . Para visualizar isso, o usuário poderá também modificar as diferenças temporais entre $K_1 \rightarrow K_3$ e $K_2 \rightarrow K_3$, deixando a matriz de topologia assim:

```
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;
1 3 sw0 tBD-5 tbN; 2 3 sw0 tBD-5 0;
4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0];
```

Os resultados são mostrados na figura 6.10B, onde se pode ver que a frequência de K_3 é o dobro de K_1 ou de K_2 . Isso ocorre em função de K_1 provocar um disparo de K_3 e, após um período tBD ms, K_2 também provocar o disparo em K_3 . Em outras palavras, K_1 OU K_2 disparam K_3 . Estas simulações demonstram duas das funções lógicas pseudo-booleanas. Há ainda que considerar o papel do pulso de *reset* global, uma rajada aplicada a ~250 ms, e que faz com que todas as assembleias fiquem silentes. A explicação é que, ao inibir K_1 e/ou K_2 , o grupo neural K_4 desarma a assembleia biestável K_1 - K_2 que se auto-sustentava. Isso equivale a dizer que o evento K_4 é uma coalizão que executa a função pseudo-booleana *NÃO*. Já o evento K_3 não é biestável, mas é uma assembleia que ocorre em função do grupo neural K_1 e K_2 reverberar. Com a dissolução da coalizão biestável, K_3 desaparece.

Outro aspecto a ser verificado diz respeito ao mesmo neuroide poder participar de mais de uma assembleia. Modifique o tempo $tsim=300$, o número de assembleias $numAssb=6$, com $nNa=5$, e modificar a topologia para:

```
Topol=[0 1 sw0 tBD tbN; 0 4 sw0 tBD/4 tbN;
1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;
1 4 sw0 tBD tbN; 4 1 sw0 tBD tbN;
1 3 sw0/2 1.5*tBD tbN; 2 3 sw0/2 0.5*tBD 0;
5 1 -sw0 1 0; 5 1 -sw0 7 0; 5 1 -sw0 36 0; 5 1 -sw0 43 0;
5 2 -sw0 19 0; 5 2 -sw0 26 0;
5 4 -sw0 19 0; 5 4 -sw0 26 0;];
```

Note que foi introduzida uma assembleia K_4 na rede que também é disparada pela assembleia K_0 num tempo $t_{BD}/4$ ms após o evento de estímulo. Note também que esta assembleia está conectada com K_1 e K_1 também está conectada com K_4 (linha 3 da matriz *Topol*[*l*]). Elas formam uma coalizão biestável. O resultado da simulação pode ser visto na figura 6.11.

Note que K_0 provoca o disparo de K_4 e algum tempo depois o disparo de K_1 . Quando K_4 dispara, esta assembleia envia pulsos para K_1 . Devido à relação temporal $K_4 \rightarrow K_1$, o pulso de K_4 faz K_1 disparar um pouco depois do pulso proveniente de K_0 .

Figura 6.11 **Compartilhamento de neuroides e interferência entre assembleias.** No gráfico, tanto K_4 quanto K_1 são disparados pelo evento de estímulo. Duas assembleias biestáveis se formam: K_1 - K_2 e K_1 - K_4 , demonstrando que os neuroides de K_1 podem participar de mais de uma assembleia. Porém, se todos os neuroides de K_1 participam em ambas as coalizões, ocorre mútua interferência K_2 e K_4 .

A partir desse momento, K_1 passa a compartilhar disparos com a assembleia biestável que forma com K_4 e também com a assembleia biestável que faz com K_2 . Um efeito colateral surge em virtude da conexão total entre assembleias *synfire*: o pulso proveniente de K_4 passa a fazer parte da assembleia biestável K_1 - K_2 , assim como o pulso que deveria ser restrito a K_1 - K_2 passa a fazer parte de K_1 - K_4 . Esse fenômeno tem menor probabilidade de acontecer quando: (1) consideramos grupos policronizados em vez de *synfire*, e (2) quando apenas um neuroide de K_1 for compartilhado com K_4 . Neste caso, K_4 deve cooptar outros “neuroides parceiros” em outras assembleias.

Demonstrar que é possível o compartilhamento individual de neuroide exigiria várias alterações no programa acima, extendendo muito a explicação. Isso pode ser inferido, apesar de já termos testado esta situação.

Observe que o peso sináptico de um neuroide isolado de K_1 não será suficiente para disparar toda a assembleia K_4 , nem K_2 . Assim, o pulso de K_4 compartilhado por um único neuroide de K_1 não dispara K_2 , desaparecendo a influência de K_4 na coalizão biestável K_1 - K_2 . Por outro lado, como o pulso um neurônio isolado não é suficiente para disparar K_4 , a influência da assembleia biestável K_1 - K_2 sobre K_4 também desaparece. Na verdade, deixa de existir a assembleia biestável K_1 - K_4 , e K_4 formará uma coalizão biestável com outro grupo de neuroides; grupo este que terá como integrante aquele “neuroide isolado” de K_1 .

A partir do código fonte do APÊNDICE ‘D’ o usuário pode alterar o modo de simulação para $simMod=1$, o número de neuroides por assembleia para $nNa=16$ e o *global reset* $tGReset=240$, reeditando a topologia da rede para os parâmetros seguintes:

```
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;
1 3 sw0 0.75*tBD tbN; 2 3 sw0 1.25*tBD 0;
4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0];
```

Fazendo esse procedimento e executando o program o usuário forçará o programa a executar com estímulos policronizados em vez de *synfire chain*.

Figura 6.12 **Estímulo por grupo policronizado.** Mudando o modo de simulação é possível fazer com que o programa gere sinais policronizados para excitação da rede. Quanto maior o número de neuroides por assembleia maior a probabilidade de um evento policronizado disparar as assembleias, revelando assim a natureza estocástica das operações com assembleias.

Note que a relação temporal entre K_0 e K_1 não foram modificadas, assim como as demais relações temporais na rede. Isso quer dizer que, embora a entrada neste modo de simulação seja policronizada, internamente a rede continua a funcionar como *synfire*. Isso quer dizer que nem sempre o padrão de disparo do grupo policronizado K_0 fará com que o grupo K_1 dispare. Para um número grande de neuroides por assembleia a probabilidade de K_1 acontecer é alta; porém, com o número de neuroides baixo pode ocorrer que nem sempre a assembleia biestável K_1 - K_2 ocorrerá. A imagem 6.12 mostra um gráfico da simulação.

O leitor é encorajado a modificar o número de neuroides por assembleias, bem como os pesos sinápticos e observar o que ocorre. Muitas vezes, o programa disparará apenas o grupo K_0 , que é um sinal forçado. Com 16 neuroides, mostrado na imagem 6.12, a probabilidade de K_1 ser disparado pelo evento K_0 cresce. Em simulações com poucos neuroides é comum o disparo de K_1 falhar e com isso todas as assembleias que dela ramificam. Falhas no disparo de K_1 podem ser explicadas porque, sendo constante o tempo de propagação entre K_0 e K_1 , é necessário que vários neuroides de K_0 disparem numa janela de tempo restrita para conseguirem disparar K_1 . Quando o padrão policronizado de K_0 é muito esparsos, a probabilidade de K_1 disparar diminui porque os pulsos chegam desencontrados em K_1 .

Uma vez disparada a assembleia K_1 a assembleia biestável com K_2 se formará, assim como a função *OU/E* executada para disparar o evento K_3 . Isso demonstra a **natureza estocástica** das funções pseudo-booleanas realizadas pelas coalizões neurais. Isso revela ainda a **hierarquia** e as **relações causais** que se estabelecem nas operações de computação por coalizões.

O programa do APÊNDICE 'E' é uma modificação para operar com todas as assembleias interconectadas como grupos policronizados. É mais difícil observar o resultado, especialmente na escala temporal que trabalhamos até o momento. Por isso o programa foi modificado para trabalhar com atrasos de 40-50 ms na propagação de spikes entre assembleias, podendo ser ajustado para mais. Com escala de tempos menores as assembleias policronizadas provavelmente falham.

A figura 6.13 mostra um algoritmo no qual o evento de entrada é K_0 , que dispara K_1 e depois dispara K_2 . As assembleias K_2 - K_3 formam um grupo policronizado. Ambos

acionam a assembleia K_4 por uma função *OU*, e a assembleia K_5 por uma função *AND*, como já foi discutido anteriormente com as assembleias synfire. O leitor notará que os grupos deixam de ser sincronizados e formam grupos esparsos, uma vez que os grupos policronizados não operam sincronamente, mas *temporalmente travados*.

Figura 6.13 **Assembleias de grupos policronizados.** No apêndice ‘E’, modo *simMod*=1, as assembleias são todas policronizadas. Note que cada assembleia tem uma forma particular de disparo que permanece nas formações seguintes, como descrito no tópico 4.4.1 (ver fig 4.3).

Observe que a forma particular de disparo de cada assembleia é repetida toda vez que a coalizão acontece, como foi previsto em teoria no tópico 4.4.1 (ver figura 4.3). Essa é uma forma mais robusta de executar as funções, ao mesmo tempo, mostra a característica mais estatística que nos grupos synfire, além de parecer mais próximo ao que ocorre biologicamente. O leitor poderá aumentar o número de neuroides por assembleias e observar que o que prevalece são “blocos” de neuroides disparando agregados.

Uma coisa a destacar é que é possível trocar o modelo de neuroide para quase todos os 23 modelos do programa (não funcionam os tipos: 7, 12, 16, 18, 19, 20 – ver figura 2.1). Quando mudamos para grupos policronizados, vamos observar que o “algoritmo” proposto continuará a ser executadas para a maioria dos modelos de neuroides. Isso demonstra que o “tipo” do neuroide pode modificar, mas a funcionalidade dentro de um ambiente de assembleias é mantida; demonstrando que a computação por assembleias depende pouco do tipo de neurônio envolvido com as operações que elas executam.

Como a escala temporal foi modificada, agora o programa também tem que varrer um grupo mais amplo de disparos acontecidos no passado. Aumentar a escala temporal significa aumentar o custo computacional.

Nas simulações descritas anteriormente foram usadas cadeias *synfire* com assembleias síncronas por serem mais fáceis de ser modeladas e visualizadas. Foram apresentadas operações com grupos policronizados somente ao final por serem menos intuitivos e sua visualização é mais complexa. O programa do APÊNDICE 'E' é similar ao anterior, e funcionalmente semelhante.

RELEVÂNCIA PARA A TESE:

Esta simulação comprova 4 hipóteses levantadas (Cf. figs. 6.12 e 6.13).

1. Grupos neurais, síncronos ou não, são capazes de auto-sustentar um *loop biestável*, desde que haja consistência temporal entre as propagações de seus pulsos e pesos sinápticos que faz a conexão entre eles.
2. A função *E* lógica é executada na coincidência temporal de pulsos de dois ou mais grupos neurais, quando os pesos das conexões sinápticas de um grupo isolado não são suficientes para gerarem o evento pós-sináptico sozinho.
3. A função *OU* lógica é executada quando os pulsos de duas ou mais assembleias atingem outra assembleia, e os pesos sinápticos de qualquer das assembleias pré- são suficientes para gerar o evento pós-sináptico.
4. Um grupo de neuroides inibitórios é capaz de executar a função *NÃO* lógica, quando desarma ou inibe as operações de outros grupos.

Melhorias podem ser introduzidas nesses programas, como uma interface mais amigável com o usuário e facilidades para implantar relações temporais individuais entre neuroides. Estes são programas protótipos, feitos em Matlab. Importante é obter e aperfeiçoar os algoritmos, para escrevê-los em linguagem de melhor desempenho, como C++, especialmente se associada a linguagens que permitam aceleração de processamento, usando computação paralela, já que a simulação de redes com número maior de assembleias demanda grande poder computacional, sendo um caminho o processamento paralelo em GPUs, por exemplo.

6.6 Resumo do capítulo

Neste capítulo foram mostradas as simulações executadas para comprovar ou validar as hipóteses levantadas nesta tese e para compreender melhor alguns aspectos da computação por meio de assembleias.

Alguns aspectos são comuns em todas as simulações, por exemplo, o modelo neural escolhido, as formas como deixamos parametrizados os aspectos mais usualmente modificados nas simulações. Isso foi mostrado nas discussões sobre o ambiente de simulação. Depois, mostramos um estudo e um programa que ajuda a escolher os neuroides que farão transdução de sinais, os “neuroides sensores”.

Na revisão sobre um artigo submetido a um congresso tempos atrás, um *reviewer* anônimo comentou que é preciso simular e realizar análises matemáticas das simulações para todas as possibilidades e hipóteses levantadas, *porque o demônio geralmente está nos detalhes*²⁹. Parece que o revisor tem razão. Um primeiro passo no sentido de testar em conjunto a abordagem da computação por assembleias foi simular um circuito que comprovasse todas as propriedades simultaneamente. Criamos um circuito contador binário e acumulador.

Num primeiro momento usamos redes sem atrasos temporais entre neuroides e obtivemos sucesso com os contadores, como foi descrito no tópico 6.3. O programa do apêndice B nesta tese permite ao usuário repetir esta simulação.

O passo seguinte foi inserir atrasos temporais na rede; o diabo está mesmo nos detalhes. O mesmo circuito contador passa a funcionar em situações extremamente rígidas de temporalidades, ou não funcionar de forma alguma. Isso poderá ser atestado pelo programa do apêndice C.

Mais importante é que novos problemas foram revelados nesse estágio das simulações e, nesse sentido, dirigiu a pesquisa para outro rumo: o estudo das relações basais entre assembleias, as execuções de funções pseudo-booleanas, as

²⁹ The devil is often in the detail, mathematical and simulation analysis may reveal a number of issues not conceived in the mere conceptual framework. Simulations may also address the issue of how robust the proposed mechanisms are in noisy situations, or how much fine-tuning and tweaking is necessary to make them work (de um reviewer anônimo).

formações e estabilidades de assembleias biestáveis, o compartilhamento de neuroides entre assembleias, os grupos policronizados, etc. Isso nos levou a construir uma ferramenta para simulações basais para coalizões, apresentada nos apêndices D e E, e discutida no tópico 6.5 acima.

Importante é que as simulações com essa ferramenta mostram que as funções que servem de base para a abordagem da computação por assembleias, assim como algumas propriedades inferidas teoricamente nos capítulos anteriores, podem ser comprovadas com o uso dessa ferramenta, desse pequeno programa em Matlab.

Dessa forma, pudemos comprovar as hipóteses mais relevantes levantadas nesta tese e, apesar de haver muito a ser investigado e simulado em termos práticos, as fundações da abordagem sobre computação por assembleias estão estabelecidas e podem ser testadas nos programas apresentados neste capítulo.

7 CONCLUSÕES

//

Esta tese tem por propósito mostrar que pode haver uma nova forma de computação em redes neurais pulsadas que se baseia no comportamento “social” dos neurônios ou neuroides quando se agrupam e formam assembleias.

Quando advogamos ser esta uma nova forma de computação, o falamos porque não encontramos na literatura abordagem semelhante. Há um consenso crescente sobre neurônios computarem por meio da temporização dos pulsos e que as redes neurais pulsadas, as de terceira geração [24], são o futuro da neurocomputação simulada por meio de computadores. Contudo, a abordagem mais usada atualmente em RNPs, a *reservoir computing* (ver [9], [22], [108], [110]) é ainda uma abordagem apoiada na ideia connexionista de que (1) basta criar um reservatório, um grupo de neuroides aleatoriamente interconectados, (2) neles injetar dados e (3) basta treinar alguns neurônios em um *layer* de saída. O resultado é que toda a computação será executada na propagação dos sinais pelo reservatório.

A abordagem aqui apresentada é diferente. Apostamos que pode haver computação na formação efêmera de grupos neurais, na forma de coalizão ou de eventos que acontecem em grupos síncronos e policronizados. A proposta, em si, não é nova, e nem há novidade em afirmar que coalizões realizam computação. Desde antes de Hebb se fala em neurônios se agrupando em assembleias. Resolvemos apostar que é possível esclarecer *como* certo tipo de computação acontece quando ocorre o fenômeno das assembleias, ou quando as coalizões interagem entre si. Para isso, levantamos algumas hipóteses e as hipóteses principais, sintetizadas no tópico 1.6, foram devidamente demonstradas.

Primeiro, supomos que neurônios utilizam a temporização para computar. Na simulação de assembleias, a temporização mostrou que pode ser um problema, ou uma solução. Um problema porque a introdução de atrasos nos contadores binários, por exemplo, fez com que a rede deixasse de executar sua função inicial: contar binário ou acumular; com isso a rede perdeu sua funcionalidade ou ficou bastante restrita. Isso comprova nossa segunda hipótese, de que não é possível negligenciar atrasos em simulações de redes neurais pulsadas. Por outro lado, as relações

temporais podem ser as responsáveis, junto com os pesos sinápticos, pelo surgimento das assembleias. Isso fica demonstrado pelos exemplos nas simulações do tópico 6.5.

Na segunda hipótese assumimos que, implícito na formação de assembleias, está a execução de alguma função lógica, cuja natureza é estocástica. A comprovação dessa hipótese pode ser obtida pelo exemplo mostrado nas figuras 6.9 e 6.10, além da experiência do contador binário. Assembleias se formam *porque* funções lógicas pseudo-booleanas *E*, *OU* e *NÃO* acontecem durante a combinação espaço-temporal dos *spikes* envolvidos na coalizão. As assembleias são *consequência* de uma dessas operações computacionais. Chamamos “funções lógicas pseudo-booleanas” porque não é necessário que *todos* os neurônios ou neuroides envolvidos na formação do evento realmente tenham pulsado e contribuído para sua formação. Na verdade, a maioria deles disparando em uma janela temporal estrita pode ser responsável pela formação de uma coalizão. Isso fica ilustrado no exemplo do tópico 6.5, em que trocamos o modo de operação (do programa do apêndice D) para grupos policronizados. A formação de assembleias começa a falhar, revelando a natureza estatística das formações, das ramificações e das dissoluções das assembleias. No referido experimento, quando um número suficiente de neuroides coindide temporalmente é que as assembleias se formam.

Também assumimos a hipótese de que assembleias podem se conectar por laços de retro-alimentação e formar coalizões biestáveis, que além de memorizarem informação representam internamente o evento que as gerou. Isso foi demonstrado desde a experiência com o contador binário, e reforçado depois da introdução dos atrasos entre assembleias. A biestabilidade das assembleias é fator importante na composição de algoritmos. Como foi mostrado que as coalizões podem fazer *loops* biestáveis e que podem executar funções lógicas ao se formarem, então mostramos por meio das simulações e exemplos do capítulo VI, que todos os elementos para criar algoritmos podem ser implantados por meio da abordagem de computação por assembleias.

Fica evidente nas simulações que neuroides podem participar de outras assembleias durante os intervalos entre pulsos; e que o peso sináptico entre os neuroides de

diferentes grupos pode garantir que um neuroide participe de outra assembleia sem interferir na coalizão em que estiver sendo analisado.

Alguém pode questionar que estamos “desenhando” a rede e as assembleias. Neste estado da investigação, sim. As redes testadas até o momento são determinísticas. Era preciso comprovar que os elementos básicos da formação de assembleias são de fato válidos e possíveis de serem computados. E é necessário primeiramente entender as relações entre todos estes componentes. Foi essa a proposta dessa tese, e foram satisfatórios os resultados obtidos.

Em defesa da postura determinística de “desenhar” as assembleias, de forçar as relações temporais e de moldar os acoplamentos sinápticos entre os neuroides da rede, podemos responder com o seguinte argumento: é aceitável a explicação de que um órgão surge em um organismo biológico porque está determinado nos genes desse organismo. E um órgão pode ser extremamente complexo, construído com grupos de células e tecidos ocupando lugares-chave, com formações de proteínas em momentos adequados, e vários outros fatores. Tudo isso cabe numa explicação plausível de que está sequenciado nos genes do organismo.

Aceitamos inclusive que existem vários centros geradores de padrões de disparos neurais com os quais um organismo nasce ou se desenvolve, porque está determinado nos genes desse organismo. Portanto, é perfeitamente aceitável “desenhar” algumas estruturas que executem computação por assembleias porque a natureza também deve lançar mão desse expediente. É possível que alguns SNs e encéfalos tenham estruturas que facilitem e guiem a formação de assembleias.

Dessa forma, temos uma visão oposta daqueles que pensam que emergência é o princípio pelo qual as redes neurais construíram encéfalos. É possível pensar em emergência no sentido de que neurônios individuais não “sabem” do resultado global da computação executada no cérebro. Nesse sentido, há um fenômeno emergente no cérebro. Mas a emergência advinda de uma abordagem puramente connexionista tem probabilidade infinitamente pequena de acontecer. Ou seja, entendemos que existe a chance de se obter formação de assembleias (ou grupos policronizados) em rede esparsa e randomicamente conectada, como provaram Izhikevich, Edelman e colaboradores [46], [90], [91]. Em organismos tais formações de assembleias

começam a ser identificadas [124]. O que não acreditamos é que seja possível obter as relações causais e hierárquicas que estamos propondo como fundamentais para a computação por assembleias simplesmente por acaso. Acreditamos que não adianta esperar que conexões aleatórias, de forma emergente, gerem o tipo de estrutura descrito nesta tese, porque a chance de se obter tal intento será infinitesimal. Não acreditamos que uma rede com inúmeros neuroides conectados aleatoriamente, por mais que sejam estrategicamente estimulados, gere algum tipo de computação parecida com o pensamento.

Em nossa visão, pensamento, cognição e qualquer outra computação sofisticada que acontece nos encéfalos não é resultado de um fenômeno puramente emergente. É resultado de uma construção. Uma construção iniciada pela natureza há milhões de anos, criando estruturas básicas que, passo a passo, permitiram aos cérebros realizar computação cada vez mais sofisticada.

Um dos recursos que os sistemas nervosos provavelmente utilizam é a formação de assembleias sobre estruturas pré-construídas. Assim, o caminho natural para iniciar as investigações sobre computação por coalizões é desenhando as primeiras estruturas de rede, e não esperando que elas venham a emergir de simulações de milhares de neuroides conectados aleatoriamente.

Conseguimos demonstrar as hipóteses principais propostas para uma abordagem de computação por assembleias. Demonstramos que é possível criar novas formas de computar usando neuroides pulsados e começamos a entender as relações entre componentes envolvidos nesse novo tipo de computação.

Mais que isso, começamos a criar ferramentas e compreender algoritmos para facilitar a implantação de computação por assembleias em novos projetos. Estamos apenas começando esta abordagem. Ela é nova e, como todo assunto que é novo, nos coloca frente a frente com as primeiras barreiras e dificuldades. Tudo indica que o fato de ter demonstrado que computação por assembleias é um novo e provavelmente frutífero caminho a ser seguido dentro de redes neurais é o grande êxito dessa tese.

7.1 Dos objetivos e resultados

O principal objetivo era demonstrar que esta é uma nova forma de pensar operações computacionais em redes neurais pulsadas. Esse objetivo foi alcançado.

Comprovamos que as assembleias se formam e se dissolvem da forma como foi previsto em teoria. Os resultados dessa comprovação ainda não apresentam resultados práticos; porém, começa a aflorar interesse pelo assunto, inclusive com repercussão internacional.

O programa protótipo apresentado aqui (apêndice D) é uma ferramenta que deve ser melhorada, e que se revela básica e poderosa, na qual (e com a qual) podemos aprender muito sobre as infinitas possibilidades advindas da interconexão de 23 tipos de neuroides, agregados em quantidades ajustáveis, com pesos sinápticos também ajustáveis e com diversas formas de relação temporal.

Os resultados são animadores porque agora sabemos identificar os principais problemas a serem enfrentados na sequência e começamos a entender processos pelos quais podemos conseguir bons resultados.

7.2 Perspectivas e continuidade do trabalho

As perspectivas da computação por assembleias são as melhores possíveis. Investigações levadas a cabo em institutos de neurociência³⁰ em todo o mundo têm mostrado que assembleias parecem ser uma das formas como SNs biológicos computam. Em vários institutos, cientistas têm tentado descobrir meios de identificar assembleias e coalizões em grupos de neurônios (Cf. [124], [127], [135]). Iniciamos uma abordagem no sentido inverso, na qual criamos os circuitos que executam algoritmos computacionais por meio da formação e dissolução de assembleias. Há muito que investigar em nossa abordagem. Podemos dizer que uma avenida foi aberta e que é possível seguir em várias direções.

³⁰ Por exemplo, o Instituto de Neurociência de Natal, página na internet: <http://www.natalneuro.org.br/> (acessada em 14/10/2011).

Primeiramente, devemos introduzir mecanismos de plasticidade e entender suas relações no curto e no longo prazo com a formação e com a estabilidade das assembleias, e isso parece ser uma necessidade mais imediata.

Será preciso ainda compreender como acontecem ajustes e aprendizado temporal nas assembleias, embora a proposta apresentada nesta tese (Cf. tópico 3.7.1) prometa ser uma boa estratégia neste sentido.

Será também necessário descobrir formas e propor algoritmos de aprendizado para a computação por assembleias como um todo. Pode ser que a hipótese de Yuste e colaboradores [120], de que o neocórtex evoluiu a partir de CPGs, seja bastante inspiradora. Criar CPGs básicos que formem estruturas capazes de aprender a formar novas assembleias, ou aprender a controlar a formação de assembleias, parece ser uma ideia com boas perspectivas.

A computação por assembleias é na essência grande consumidora de recursos computacionais. Além disso, é o tipo de computação que deve acontecer com estímulo contínuo e ser computada em tempo real. Portanto, desenvolver algoritmos que aumentem o desempenho computacional é imperativo e é outro desdobramento para investigações que pode ter origem a partir das propostas aqui apresentadas.

Os conhecimentos adquiridos a partir dessa abordagem estão mais próximos do mundo biológico, da mesma forma que conhecimentos advindos da biologia e neurociência podem ser mais facilmente simulados na abordagem que aqui se propõe. Não estamos com isso advogando que assembleias seja “a forma” que os SNs biológicos computam, mas uma das formas, que possivelmente acontece, em algum grau, em sistemas nervosos naturais.

Assim, a perspectiva de aproximação de conhecimento nos dois sentidos, da biologia para a engenharia e vice-versa, é muito boa dentro da abordagem da computação por coalizões. Podemos fazer uso dos programas de simulação de assembleias pulsadas em estudos de formações de memória, de circuitos de atenção, de geração de padrões de disparo, de controle motor, enfim, uma gama enorme de utilizações que podem ser úteis em várias instâncias para biólogos.

Existe atualmente uma tecnologia relativamente barata de simulação na qual é possível simular mundos virtuais com física, colisões, percepção visual, simulação

de sensores diversos e vários aspectos e propriedades dos mundos reais. Trata-se dos motores de jogos (*game engines*), softwares que estão disponíveis atualmente em diversos formatos, inclusive em código aberto. Este pode ser um ambiente ideal para simulação de agentes virtuais controlados por assembleias neurais pulsadas.

Seria possível, por exemplo, criar robôs que imitem insetos, bípedes, quadrúpedes, etc; que podem ser dotados de visão e de outros sistemas sensoriais. Tais agentes seriam estimulados o tempo todo dentro dos motores de jogos, uma vez que a física simulada no ambiente reflete sobre todos os corpos presentes na simulação. Dessa forma, os agentes teriam que aprender a responder aos estímulos do meio. Portanto, este ambiente representa uma boa opção para a continuidade dessa abordagem sobre computação por meio de assembleias de neuroides pulsados.

Observação final – edição revisada: versões atualizadas do programa de simulação dos fundamentos de “computação por assembleias neurais” (Neural Assembly Computing), bem como um pequeno tutorial, estão disponíveis e serão mantidos para download em:

http://www.neuralassembly.org/download/NAC_Fundamentals.m

http://www.neuralassembly.org/download/Tutorial_NAC_Fundamentals.pdf

(acesso em 05/ 01/2012).

REFERÊNCIAS

//

- [1] DEL-NERO, H. S. *O sítio da mente: pensamento, emoção e vontade no cérebro humano*. São Paulo: Collegium Cognitio, 1997.
- [2] DEL-NERO, H. S. *Computação topológica e controle voluntário em arquiteturas naturais e artificiais*. 1997. Tese de doutoramento Dr. Del Nero, orientação: Dr. J. R. C. Piqueira.
- [3] KANDEL, E. R.; SCHWARTZ, J. H.; JESSEL, T. M. *Principles of neural science*. 4th ed. New York, NY: McGrall-Hill Health Prof. Division, 2000.
- [4] ARBIB, M. A. *The handbook of brain theory and neural networks*. 2nd ed. Cambridge, MA: MIT Press, 2003.
- [5] PURVES, D. et al. *Neuroscience*. 3rd. ed. Sunderland, MA: Sinauer Associates, Inc., 2004.
- [6] BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. *Neurociências: desvendando o sistema nervoso*. 3a. ed. Porto Alegre, RS, Brasil: Artmed Editora S.A., 2008.
- [7] KOCH, C. *Biophysics of computation: information processing in single neurons*. New York, NY: Oxford University Press, 1999.
- [8] HAYKIN, S. S. *Neural networks: a comprehensive foundation*. 2nd edition. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [9] SCHRAUWEN, B.; VERSTRAETEN, D.; VAN CAMPENHOUT, J. An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks ESANN'2007. Editor VERLEYSEN, M. p. 471–482.
- [10] GROSS, C. G. Genealogy of the grandmother cell. *The Neuroscientist*, v. 8, n. 5, p. 512–518, 2002.
- [11] QUIROGA, R. Q.; REDDY, L.; KREIMAN, G.; KOCH, C.; FRIED, I. Invariant visual representation by single neurons in the human brain. *Nature*, London, v. 435, n. 7045, p. 1102–1107, 2005.
- [12] QUIROGA, R. Q.; MUKAMEL, R.; ISHAM, E. A.; MALACH, R.; FRIED, I. Human single-neuron responses at the threshold of conscious recognition. *PNAS - Proceedings of the National Academy of Sciences*, v. 105, n. 9, p. 3599–3604, 2008.

- [13] CONNOR, C. E. Neuroscience friends and grandmothers. *Nature*, London, v. 435, n. 7045, p. 1036–1037, 2005.
- [14] QUIROGA, R. Q.; KREIMAN, G.; KOCH, C.; FRIED, I. Sparse but not 'grandmother-cell' coding in the medial temporal lobe. *Trends in Cognitive Sciences*, v. 12, n. 3, p. 87–91, 2008.
- [15] PIQUEIRA, J.; ORSATTI, F.; MONTEIRO, L. Computing with phase locked loops: choosing gains. *Neural Networks, IEEE Transactions on*, v. 14, n. 1, p. 243–247, 2003.
- [16] MONTEIRO, L.; CANTO, N.; CHAUI-BERLINCK, J.; ORSATTI, F.; PIQUEIRA, J. Global and partial synchronism in phase-locked loop networks. *Neural Networks, IEEE Transactions on*, v. 14, n. 6, p. 1572–1575, 2003.
- [17] ORSATTI, F. M.; CARARETO, R.; PIQUEIRA, J. Multiple synchronous states in static delay-free mutually connected pll networks. *Signal Processing*, v. 90, n. 6, p. 2072–2082, 2010.
- [18] DEL-MORAL-HERNANDEZ, E. Recursive nodes with rich dynamics as modeling tools for cognitive functions. In: PERLOVSKY, L.; KOZMA, R. (Eds.) *Neurodynamics of Cognition and Consciousness*. Springer Berlin / Heidelberg, 2007. v. 25 of *Understanding Complex Systems*, p. 279–304.
- [19] DEL-MORAL-HERNANDEZ, E.; SANDMANN, H.; ARAUJO, G. Context dependent pattern recognition: a framework for hybrid architectures bridging chaotic neural networks based on recursive processing elements and symbolic information. In: . Editor PRESS, I. IEEE Press, c2009. v. 1 of *IJCNN'09*. p. 663–670.
- [20] NETTO, M. L. Computação evolutiva e cognitiva - simulação em vida artificial e cognição, 2007. Tese de livre-docência. Escola Politécnica da Universidade de S. Paulo, Depto. Eng. Eletrônica.
- [21] GRAMSS, T.; GROSS, M.; MITCHELL, M.; PELLIZZARI, T.; BORNHOLDT, S. *Non-standard computation: Molecular computation - cellular automata - evolutionary algorithms - quantum computers*. Weinheim: Wiley-VCH, 1998.
- [22] PAUGAM-MOISY, H.; BOHTE, S. Computing with spiking neuron networks. In: ROZENBERG, G.; BÄCK, T.; KOK, J. N. (Eds.) *Handbook of Natural Computing. 1st ed.* Heidelberg, Germany: Springer-Verlag, 2010. v. 1, p. 1–47.
- [23] BRETTE, R. et al. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, v. 23, n. 3, p. 349–398, 2007.

- [24] MAASS, W. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, v. 10, n. 9, p. 1659–1671, 1997.
- [25] IZHIKEVICH, E. M. *Dynamical systems in neuroscience: the geometry of excitability and bursting*. Cambridge, Massachusetts: The MIT Press, 2007.
- [26] IZHIKEVICH, E. M. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, v. 15, n. 5, p. 1063–1070, 2004.
- [27] IZHIKEVICH, E. M. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, v. 14, n. 6, p. 1569–1572, 2003.
- [28] GERSTNER, W.; KISTLER, W. M. *Spiking neuron models: single neurons, populations, plasticity*. Cambridge, UK: Cambridge University Press, 2002.
- [29] ALVAREZ, V. A.; SABATINI, B. L. Anatomical and physiological plasticity of dendritic spines. *Annual Review of Neuroscience*, v. 30, n. 1, p. 79–97, 2007.
- [30] CHKLOVSKII, D. B.; MEL, B. W.; SVOBODA, K. Cortical rewiring and information storage. *Nature*, London, v. 431, n. 7010, p. 782–788, 2004.
- [31] DE-PAOLA, V. et al. Cell type-specific structural plasticity of axonal branches and boutons in the adult neocortex. *Neuron*, v. 49, n. 6, p. 861–875, 2006.
- [32] TURRIGIANO, G. G.; NELSON, S. B. Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience*, v. 5, n. 2, p. 97–107, 2004.
- [33] POZO, K.; GODA, Y. Unraveling mechanisms of homeostatic synaptic plasticity. *Neuron*, v. 66, n. 3, p. 337–351, 2010.
- [34] NELSON, S. B.; TURRIGIANO, G. G. Strength through diversity. *Neuron*, v. 60, n. 3, p. 477–482, 2008.
- [35] BI, G.-Q.; POO, M.-M. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, v. 24, n. 1, p. 139–166, 2001.
- [36] ABBOTT, L. F. AND NELSON, S. B. Synaptic plasticity: taming the beast. *Nature Neuroscience*, v. 3, p. 1178–1183, 2000.
- [37] MARKRAM, H.; LÜBKE, J.; FROTSCHER, M.; SAKMANN, B. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, Washington, v. 275, n. 5297, p. 213–215, 1997.

[38] SJÖSTRÖM, P. J.; RANCZ, E. A.; ROTH, A.; HÄUSSER, M. Dendritic excitability and synaptic plasticity. *Physiological Reviews*, v. 88, n. 2, p. 769–840, 2008.

[39] MORRISON, A.; DIESMANN, M.; GERSTNER, W. Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics*, v. 98, n. 6, p. 459–478, 2008.

[40] DESTEXHE, A.; MARDER, E. Plasticity in single neuron and circuit computations. *Nature*, London, v. 431, n. 7010, p. 789–795, 2004.

[41] LONDON, M.; HÄUSSER, M. Dendritic computation. *Annual Review of Neuroscience*, v. 28, n. 1, p. 503–532, 2005.

[42] SPRUSTON, N. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, v. 9, n. 3, p. 206–221, 2008.

[43] VETTER, P.; ROTH, A.; HÄUSSER, M. Propagation of action potentials in dendrites depends on dendritic morphology. *Journal of Neurophysiology*, v. 85, n. 2, p. 926–937, 2001.

[44] ABELES, M. Synfire chains. *Scholarpedia*, v. 4, n. 7, p. 1441, 2009. URL: <http://www.scholarpedia.org/article/Synfire_chains>. Acesso em 20 out. 2011.

[45] BIENENSTOCK, E. A model of neocortex. *Network: Computation in Neural Systems*, v. 6, n. 1, p. 179–224, 1995.

[46] IZHIKEVICH, E. M. Polychronization: computation with spikes. *Neural Computation*, v. 18, n. 2, p. 245–282, 2006.

[47] IZHIKEVICH, E. M.; HOPPENSTEADT, F. C. Polychronous wavefront computations. *International Journal Of Bifurcation And Chaos*, v. 19, n. 05, p. 1733–1739, 2009.

[48] RANHEL, J.; LIMA, C. V.; MONTEIRO, J.; KOGLER JR., J. E.; NETTO, M. L. Bistable memory and binary counters in spiking neural network. In: . Editor PRESS, I. IEEE Press, c2011. v. 1 of 2011 *IEEE Symposium on Foundations of Computational Intelligence. FOCI 2011 Proceedings*. p. 66–73.

[49] RANHEL, J.; NETTO, M. L. Computing with spiking neural assemblies. *IEEE Trans on Systems, Man, and Cybernetics-Part B*. (em revisão, ver Apêndice F).

[50] MONTEIRO, J. L. R.; KOGLER, J. A. E.; RANHEL, J. A. H.; NETTO, M. L. On building a memory evolutive system for application to learning and cognition modeling. In: HUSSAIN, A. et al. (Eds.) *Brain Inspired Cognitive Systems 2008*. New York, NY: Springer, 2010. p. 19–39.

- [51] CESSAC, B.; PAUGAM-MOISY, H.; VIÉVILLE, T. Overview of facts and issues about neural coding by spikes. *Journal of Physiology-Paris*, v. 104, n. 1-2, p. 5–18, 2010.
- [52] SÖHL, G.; MAXEINER, S.; WILLECKE, K. Expression and functions of neuronal gap junctions. *Nature Reviews Neuroscience*, v. 6, n. 3, p. 191–200, 2005.
- [53] ZHANG, H.-Y.; LI, W.-C.; HEITLER, W. J.; SILLAR, K. T. Electrical coupling synchronises spinal motoneuron activity during swimming in hatchling xenopus tadpoles. *The Journal of Physiology*, v. 587, n. 18, p. 4455–4466, 2009.
- [54] IZHIKEVICH, E. M. Solving the distal reward problem through linkage of stdp and dopamine signaling. *Cerebral Cortex*, v. 17, n. 10, p. 2443–2452, 2007.
- [55] ROJAS, R. *Neural networks: a systematic introduction*. Berlin: Springer, 1996.
- [56] TOUBOUL, J.; BRETTE, R. Dynamics and bifurcations of the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, v. 99, n. 4, p. 319–334, 2008.
- [57] CESSAC, B.; VIEVILLE, T. On dynamics of integrate-and-fire neural networks with conductance based synapses. *Frontiers in Computational Neuroscience*, v. 5, n. 0, p. 1–20, 2008.
- [58] DAYAN, P.; ABBOTT, L. F. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Cambridge, MA: The MIT Press, 2005.
- [59] CRUSE, H. *Neural networks as cybernetic systems. 3rd and revised edition*. Bielefeld, Germany: Brains, Minds & Media, 2009.
- [60] IZHIKEVICH, E. M.; HOPPENSTEADT, F. C. Classification of bursting mapping. *International Journal of Bifurcation and Chaos*, v. 14, n. 11, p. 3847–3854, 2004.
- [61] LUGER, G. F. *Inteligência artificial: estruturas e estratégias para a solução de problemas complexos. 4a. ed.* Porto Alegre - Brasil: Bookmann, 2004.
- [62] THORPE, S.; DELORME, A.; VANRULLEN, R. Spike-based strategies for rapid processing. *Neural Networks*, v. 14, n. 6-7, p. 715–725, 2001.
- [63] PERRETT, D. I.; ROLLS, E. T.; CAAN, W. Visual neurones responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, v. 47, n. 3, p. 329–342, 1982.
- [64] THORPE, S.; FIZE, D.; MARLOT, C. Speed of processing in the human visual system. *Nature*, London, v. 381, n. 6582, p. 520–522, 1996.

- [65] ROLLS, E. T.; TOVEE, M. J. Processing speed in the cerebral cortex and the neurophysiology of visual masking. *Proceedings: Biological Sciences*, v. 257, n. 1348, p. 9–15, 1994.
- [66] ROLLS, E. T.; FRANCO, L.; AGGELOPOULOS, N. C.; JEREZ, J. M. Information in the first spike, the order of spikes, and the number of spikes provided by neurons in the inferior temporal visual cortex. *Vision Research*, v. 46, n. 25, p. 4193–4205, 2006.
- [67] MAINEN, Z.; SEJNOWSKI, T. Reliability of spike timing in neocortical neurons. *Science*, Washington, v. 268, n. 5216, p. 1503–1506, 1995.
- [68] IKEGAYA, Y. et al. Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, Washington, v. 304, n. 5670, p. 559–564, 2004.
- [69] SWADLOW, H. A. Efferent neurons and suspected interneurons in motor cortex of the awake rabbit: axonal properties, sensory receptive fields, and subthreshold synaptic inputs. *Journal of Neurophysiology*, v. 71, n. 2, p. 437–453, 1994.
- [70] WEHR, M.; LAURENT, G. Odour encoding by temporal sequences of firing in oscillating neural assemblies. *Nature*, London, v. 384, n. 6605, p. 162–166, 1996.
- [71] TIMOSZCZUK, A. P. *Reconhecimento automático do locutor com redes neurais pulsadas*. 2004. Tese (Doutorado em Física) - Escola Politécnica da Universidade de S. Paulo, Depto. Eng. Eletrônica, 2004. Tese doutoramento, boa revisão sobre RNP, orientação: Cabral Junior, Euvaldo Ferreira.
- [72] WANG, D. Emergent synchrony in locally coupled neural oscillators. *Neural Networks, IEEE Transactions on*, v. 6, n. 4, p. 941–948, 1995.
- [73] HOPPENSTEADT, F. C.; IZHIKEVICH, E. M. Synaptic organizations and dynamical properties of weakly connected neural oscillators i. analysis of a canonical model. *Biological Cybernetics*, v. 75, n. 2, p. 117–127, 1996.
- [74] FOSS, J.; MILTON, J. Multistability in recurrent neural loops arising from delay. *Journal of Neurophysiology*, v. 84, n. 2, p. 975–985, 2000.
- [75] ENGEL, A. K.; FRIES, P.; SINGER, W. Dynamic predictions: oscillations and synchrony in top-down processing. *Nature Reviews Neuroscience*, v. 2, n. 10, p. 704–716, 2001.
- [76] BUZSÁKI, G.; DRAGUHN, A. Neuronal oscillations in cortical networks. *Science*, Washington, v. 304, n. 5679, p. 1926–1929, 2004.

- [77] HOPFIELD, J. J.; BRODY, C. D. What is a moment? transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences of the United States of America*, Washington, v. 98, n. 3, p. 1282–1287, 2001.
- [78] JINDE, C.; GUANRONG, C.; PING, L. Global synchronization in an array of delayed neural networks with hybrid coupling. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, v. 38, n. 2, p. 488–498, 2008.
- [79] HUAGUANG, Z.; TIEDONG, M.; GUANG-BIN, H.; ZHILIANG, W. Robust global exponential synchronization of uncertain chaotic delayed neural networks via dual-stage impulsive control. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, v. 40, n. 3, p. 831–844, 2010.
- [80] MARTINEZ, R.; PAUGAM-MOISY, H. Algorithms for structural and dynamical polychronous groups detection. In: ALIPPI, C.; POLYCARPOU, M.; PANAYIOTOU, C.; ELLINAS, G. (Eds.) *Artificial Neural Networks – ICANN 2009*. Springer Berlin / Heidelberg, 2009. v. 5769 of *Lecture Notes in Computer Science*, p. 75–84.
- [81] PAUGAM-MOISY, H.; MARTINEZ, R.; BENGIO, S. Delay learning and polychronization for reservoir computing. *Neurocomputing*, v. 71, n. 7-9, p. 1143–1158, 2008.
- [82] VANRULLEN, R.; GUYONNEAU, R.; THORPE, S. J. Spike times make sense. *Trends in Neurosciences*, v. 28, n. 1, p. 1–4, 2005.
- [83] HODGKIN, A. L.; HUXLEY, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, v. 117, n. 4, p. 500–544, 1952.
- [84] DESTEXHE, A.; RUDOLPH, M.; PARÉ, D. The high-conductance state of neocortical neurons *in vivo*. *Nature Review Neuroscience*, v. 4, n. 0, p. 739–751, 2003.
- [85] DESTEXHE, A. Conductance-based integrate-and-fire models. *Neural Computation*, v. 9, n. 3, p. 503–514, 1997.
- [86] BRUNEL, N.; VAN ROSSUM, M. Lapique's 1907 paper: from frogs to integrate-and-fire. *Biological Cybernetics*, v. 97, n. 5, p. 337–339, 2007.
- [87] ABBOTT, L. F. Lapique's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, v. 50, n. 5-6, p. 303–304, 1999.
- [88] BRETTE, R.; GERSTNER, W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, v. 94, n. 5, p. 3637–3642, 2005.

- [89] IZHIKEVICH, E. M. Hybrid spiking models. *Phil. Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 368, n. 1930, p. 5061–5070, 2010.
- [90] IZHIKEVICH, E. M.; GALLY, J. A.; EDELMAN, G. M. Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, v. 14, n. 8, p. 933–944, 2004.
- [91] IZHIKEVICH, E. M.; EDELMAN, G. M. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, v. 105, n. 9, p. 3593–3598, 2008.
- [92] WATT, A. J.; DESAI, N. S. Homeostatic plasticity and stdp: keeping a neuron's cool in a fluctuating world. *Frontiers in Synaptic Neuroscience*, v. 2, n. 0, p. 1–16, 2010.
- [93] TURRIGIANO, G. G. The self-tuning neuron: synaptic scaling of excitatory synapses. *Cell*, v. 135, n. 3, p. 422–435, 2008.
- [94] RICH, M. M.; WENNER, P. Sensing and expressing homeostatic synaptic plasticity. *Trends in Neurosciences*, v. 30, n. 3, p. 119–125, 2007.
- [95] CAMPANAC, E.; DEBANNE, D. Plasticity of neuronal excitability : Hebbian rules beyond the synapse. *Archives italiennes de biologie*, v. 145, n. 3-4, p. 277–287, 2007.
- [96] DAVIS, G. W. Homeostatic control of neural activity: from phenomenology to molecular design. *Annual Review of Neuroscience*, v. 29, n. 1, p. 307–323, 2006.
- [97] ZHANG, W.; LINDEN, D. J. The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, v. 4, n. 11, p. 885–900, 2003.
- [98] MARKRAM, H.; WANG, Y.; TSODYKS, M. Differential signaling via the same axon of neocortical pyramidal neurons. *PNAS - Proceedings of the National Academy of Sciences*, v. 95, n. 9, p. 5323–5328, 1998.
- [99] IZHIKEVICH, E. M.; DESAI, N. S. Relating stdp to bcm. *Neural Computation*, v. 15, n. 7, p. 1511–1523, 2003.
- [100] TRIESCH, J. Synergies between intrinsic and synaptic plasticity mechanisms. *Neural Computation*, v. 19, n. 4, p. 885–909, 2007.
- [101] TOYOIZUMI, T.; PFISTER, J.-P.; AIHARA, K.; GERSTNER, W. Optimality model of unsupervised spike-timing-dependent plasticity: synaptic memory and weight distribution. *Neural Computation*, v. 19, n. 3, p. 639–671, 2007.

- [102] BARNES, S. J.; FINNERTY, G. T. Sensory experience and cortical rewiring. *The Neuroscientist*, v. 16, n. 2, p. 186–198, 2010.
- [103] HOLTMAAT, A.; SVOBODA, K. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nature Reviews Neuroscience*, v. 10, n. 9, p. 647–658, 2009.
- [104] STEPANYANTS, A.; HOF, P. R.; CHKLOVSKII, D. B. Geometry and structural plasticity of synaptic connectivity. *Neuron*, v. 34, n. 2, p. 275–288, 2002.
- [105] SJÖSTRÖM, P. J.; TURRIGIANO, G. G.; NELSON, S. B. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, v. 32, n. 6, p. 1149–1164, 2001.
- [106] NATSCHLÄGER, T.; RUF, B. Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, v. 9, n. 3, p. 319–332, 1998.
- [107] LUKOSEVICIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, v. 3, n. 3, p. 127–149, 2009.
- [108] MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, v. 14, n. 11, p. 2531–2560, 2002.
- [109] JAEGER, H. The "echo state" approach to analysing and training recurrent neural networks. *GMD Report*, v. 148, p. 1–47, 2001.
- [110] JAEGER, H.; MAASS, W.; PRINCIPE, J. Editorial: special issue on echo state networks and liquid state machines. *Neural Networks*, v. 20, n. 3, p. 287–289, 2007.
- [111] RAMIREZ, J.-M.; TRYBA, A. K.; PEÑA, F. Pacemaker neurons and neuronal networks: an integrative view. *Current Opinion in Neurobiology*, v. 14, n. 6, p. 665–674, 2004.
- [112] SATTERLIE, R. A. Reciprocal inhibition and postinhibitory rebound produce reverberation in a locomotor pattern generator. *Science*, Washington, v. 229, n. 4711, p. 402–404, 1985.
- [113] BRIGGMAN, K.; KRISTAN, W. Multifunctional pattern-generating circuits. *Annual Review of Neuroscience*, v. 31, n. 1, p. 271–294, 2008.
- [114] CANAVIER, C. C.; ACHUTHAN, S. Pulse coupled oscillators and the phase resetting curve. *Mathematical Biosciences*, v. 226, n. 2, p. 77–96, 2010.

- [115] MARDER, E. Motor pattern generation. *Current Opinion in Neurobiology*, v. 10, n. 6, p. 691–698, 2000.
- [116] GRILLNER, S. Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron*, v. 52, n. 5, p. 751–766, 2006.
- [117] GRILLNER, S. The motor infrastructure: from ion channels to neuronal networks. *Nature Review Neurosciences*, v. 4, n. 7, p. 573–586, 2003.
- [118] PRINZ, A. A.; BUCHER, D.; MARDER, E. Similar network activity from disparate circuit parameters. *Nature Neuroscience*, v. 7, n. 12, p. 1345–1352, 2004.
- [119] FELDMAN, J. L.; DEL NEGRO, C. A. Looking for inspiration: new perspectives on respiratory rhythm. *Nature Reviews Neuroscience*, v. 7, n. 3, p. 232–241, 2006.
- [120] YUSTE, R.; MACLEAN, J. N.; SMITH, J.; LANSNER, A. The cortex as a central pattern generator. *Nature Reviews Neuroscience*, v. 6, n. 6, p. 477–483, 2005.
- [121] WAXMAN, S.; BENNETT, M. V. L. Relative conduction velocities of small myelinated and non-myelinated fibres in the central nervous system. *Nature new biology*, v. 238, p. 217–219.
- [122] MA, J.; WU, J. Multistability in spiking neuron models of delayed recurrent inhibitory loops. *Neural Comput.*, v. 19, n. 8, p. 2124–2148, 2007.
- [123] ENGEL, J.; BRAGIN, A.; STABA, R.; MODY, I. High-frequency oscillations: what is normal and what is not? *Epilepsia*, v. 50, n. 4, p. 598–604, 2009.
- [124] LOPES-DOS SANTOS, V.; CONDE-OCAZONEZ, S.; NICOLELIS, M. A. L.; RIBEIRO, S. T.; TORT, A. B. L. Neuronal assembly detection and cell membership specification by principal component analysis. *PLoS ONE*, v. 6, n. 6, p. e20996, 2011.
- [125] GERSTEIN, G. L.; BEDENBAUGH, P.; AERTSEN, A. M. H. J. Neuronal assemblies. *Biomedical Engineering, IEEE Transactions on*, v. 36, n. 1, p. 4–14, 1989.
- [126] MOUNTCASTLE, V. B. Modality and topographic properties of single neurons of cat's somatic sensory cortex. *Journal of neurophysiology*, v. 20, n. 4, p. 408–434, 1957.
- [127] NICOLELIS, M.; BACCALA, L.; LIN, R.; CHAPIN, J. Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. *Science*, Washington, v. 268, n. 5215, p. 1353–1358, 1995.

- [128] GRAY, C. M.; SINGER, W. Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, Washington, v. 86, n. 5, p. 1698–1702, 1989.
- [129] KISTLER, W.; DE ZEEUW, C. Time windows and reverberating loops: a reverse-engineering approach to cerebellar function. *The Cerebellum*, v. 2, n. 1, p. 44–54, 2003.
- [130] WICKENS, J.; MILLER, R. A formalisation of the neural assembly concept 1.constraints on neural assembly size. *Biological Cybernetics*, v. 77, n. 5, p. 351–358, 1997.
- [131] MARR, D. *Vision: a computational investigation into the human representation and processing of visual information (1st. ed. 1982)*. Cambridge, MA: MIT Press, 2010.
- [132] SMITH, C. U. M. *Biology of sensory systems*. West Sussex - England: John Wiley & Sons, 2000.
- [133] SCHMIDT-RHAESA, A. *The evolution of organ systems*. New York, NY: Oxford University Press, 2007.
- [134] LAND, M. F.; NILSSON, D.-E. *Animal eyes*. New York: Oxford University Press, 2002.
- [135] HARRIS, K. D.; CSICSVARI, J.; HIRASE, H.; DRAGOI, G.; BUZSAKI, G. Organization of cell assemblies in the hippocampus. *Nature*, London, v. 424, n. 6948, p. 552–556, 2003.
- [136] GRAU, J. W. Learning and memory without a brain. In: BEKOFF, M.; ALLEN, C.; BURGHARDT, G. M. (Eds.) *The cognitive Animal: Empirical and Theoretical Perspectives on Animal Cognition*, Bradford Book. Cambridge, MA: MIT Press, 2002. p. 77–87.
- [137] ANDERSON, M. L. Embodied cognition: a field guide. *Artificial Intelligence*, Amsterdam, v. 149, n. 1, p. 91–130, 2003.
- [138] BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. *Neuroscience: Exploring the brain. 3rd ed.* Baltimore, MD: Lippincott Williams & Wilkins, 2007.
- [139] BEAN, B. P. The action potential in mammalian central neurons. *Nature Reviews Neuroscience*, v. 8, n. 6, p. 451–465, 2007.
- [140] CAULI, B. et al. Molecular and physiological diversity of cortical nonpyramidal cells. *The Journal of Neuroscience*, v. 17, n. 10, p. 3894–3906, 1997.

- [141] ALLE, H.; ROTH, A.; GEIGER, J. R. P. Energy-efficient action potentials in hippocampal mossy fibers. *Science*, Washington, v. 325, n. 5946, p. 1405–1408, 2009.
- [142] MORTON, G. J. et al. Central nervous system control of food intake and body weight. *Nature*, London, v. 443, n. 7109, p. 289–295, 2006.
- [143] VAN DEN TOP, M. et al. Orexigen-sensitive npy/agrp pacemaker neurons in the hypothalamic arcuate nucleus. *Nature Neuroscience*, v. 7, n. 5, p. 493–494, 2004.
- [144] GAO, Q.; HORVATH, T. L. Neurobiology of feeding and energy expenditure. *Annual Review of Neuroscience*, v. 30, n. 1, p. 367–398, 2007.
- [145] BEER, R. D. The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, v. 11, n. 4, p. 209–243, 2003.

APÊNDICE A

Código MATLAB para levantamento da curva $F \times I_n \times P$ dos neuroides

```
% Neuroides TRANSDUTORES IxFreq: Joao Ranhel - V.1.1 - 08/2010
% Prog. p/ levantar curvas de respostas de disparo dos neuroides (nrds)xI excitação
% Finalidade: obter curvas de respostas FxIxParâmetros de nrds do 'simple '
% Permite escolher transdutores I->Spikes e respectivas faixas de operação
% Dada In de excitação DC; certo tipo (ip) de nrd dispara numa frequência; o prog.
% calc resp Freq p/ cada In ('Imin'<= In <='Imax') com passo definido em 'step'
% Há duas regiões de destaque nas respostas dos nrds:
% (1) de 0.1 pA até 25 pA - step = 0.1 (representa: acoplamentos fracos)
% (2) de 1 pA até 100 pA - step = 0.5 (representa: acoplamentos fortes)
% Obs: pode-se levantar a curva COM ou SEM ruídos agregados - ver 'simMod'
% -----

clear; clc; % reinicia matlab vars e limpa tela
ttime=500; % Tempo da simulação (ms) para cada I aplicado ao nrd
Imin=1; % menor val de corrente Iin a excitar os nrds (pA)
step=0.5; % incremento na corrente In a cada ttime
Imax=100; % maior val de corrente Iin a excitar os nrds
simMod=2; % modo de simulação: =0 sem ruído; In é CC In(DC)
% =1 In(DC)+ruído (% do sinal In DC)
% =2 In*DC)+ruído (.05 pA randomico dist. normal).

% -- Inicializa PARAMETROS dos neuroides -- %
ip=7 % ATENÇÃO: ESCOLHER um indx para o tipo de NEUROIDE a ser testado

% Matriz com os parâmetros para os 23 classes de nrds - Izhikevich, Simple Model
Pars=[0.02 0.2 -65 6 -70 14 ;... % 1 tonic spiking
0.02 0.25 -65 6 -64 0.5 ;... % 2 phasic spiking
0.02 0.2 -50 2 -70 15 ;... % 3 tonic bursting
0.02 0.25 -55 0.05 -64 0.6 ;... % 4 phasic bursting
0.02 0.2 -55 4 -70 10 ;... % 5 mixed mode
0.01 0.2 -65 8 -70 30 ;... % 6 spike freq adaptation
0.02 -0.1 -55 6 -60 0 ;... % 7 Class 1
0.2 0.26 -65 0 -64 0 ;... % 8 Class 2
0.02 0.2 -65 6 -70 7 ;... % 9 spike latency
0.05 0.26 -60 0 -62 0 ;... % 10 subthreshold oscillations
0.1 0.26 -60 -1 -62 0 ;... % 11 resonator
0.02 -0.1 -55 6 -60 0 ;... % 12 integrator
0.03 0.25 -60 4 -64 0 ;... % 13 rebound spike
0.03 0.25 -52 0 -64 0 ;... % 14 rebound burst
0.03 0.25 -60 4 -64 0 ;... % 15 threshold variability
1 1.5 -60 0 -61 -65 ;... % 16 bistability
1 0.2 -60 -21 -70 0 ;... % 17 DAP
0.02 1 -55 4 -65 0 ;... % 18 accommodation
-0.02 -1 -60 8 -63.8 80 ;... % 19 inh-induced spiking
-0.026 -1 -45 0 -63.8 80 ;... % 20 inh-induced bursting
0.01 0.2 -62 0.5 -62 10 ;... % 21 Regular Spiking
0.1 0.2 -65 2 -65 10 ;... % 22 Fast Spiking
0.02 -0.1 -55 6 -55 0] ; % 23 Class XG

% Inicializa matrizes a serem usadas para nas equações do neuroide
nN=1; % Num total de nrds
A=Pars(ip,1)*ones(nN,1); B=Pars(ip,2)*ones(nN,1);
C=Pars(ip,3)*ones(nN,1); D=Pars(ip,4)*ones(nN,1);

% --- Loop de Simulação ---
Spks=[]; % Ini uma matriz p/ salvar 'tempo' e 'nrd' que
dispararam %
Plx=Imin:step:Imax; % vals p/ eixo X do gráfico (curva I x F)
Ply=zeros(size(Plx)); % a ser usada no gráfico
Plz=zeros(size(Plx)); % a ser usada no gráfico
y=0;
for Idc=Plx(1,:) % p/ Idc = cada valor em Plx (de Imin até Imax)
V=Pars(ip,5)*ones(nN,1); % ini matriz da tensão de membrana
In=zeros(nN,1); % ini correntes sinápticas In de cada nrd
U=B.*V; % U = var de recuperação da tensão da membrana
y=y+1; % indx para as ordenadas do gráfico (y)
cont1=0; % conta disparos de um nrd durante período 't'
cont2=0; % conta num de passos durante simulação 't'
```

```

for t=1:ttime                                     % para simulação de 1...ttime ms!
    if simMod==1
        In=Idc+0.02*Idc*randn(nN,1);             % CORRENTE DC + 2% ruído aplicado ao nrd
    elseif simMod==2
        In=Idc+0.1*randn(nN,1);                   % CORRENTE DC + ruído (+- .05pA randomico normal)
    else
        In=Idc;                                    % CORRENTE DC sem ruído
    end
    cont2=cont2+1;                                  % inc cont2
    Fired=find(V>=30);                              % listar todos os nrds que dispararam
    if ~isempty(Fired)                              % se ninguém disparou, próx t
        cont1=cont1+1;                             % se disparou, inc cont1
    end
    V(Fired,1)=C(Fired,1);                         % todos os nrds que dispararam: reset p/ val C()!
    U(Fired,1)=U(Fired,1)+D(Fired,1);              % var refração membrana é recalculada
    V=V+0.5*(0.04*V.^2+5*V+140-U+In);             % calcula tensão membrana para prox iteração
    V=V+0.5*(0.04*V.^2+5*V+140-U+In);             % passo de 0.5 ms para estabilidade numérica
    U=U+A.*(B.*V-U);                               % calcula novo val de U para prox iteração
end
Ply(y)=cont1;
Plz(y)=cont2;
end
plot(Plx,Ply,'.b-');
['ok']

```

APÊNDICE B

Código MATLAB para Memória Biestável e Contadores SEM DELAY

```
% MEM BIESTÁVEL E CONTADORES BIN em RNP SEM DELAY: João Ranhel - V.7 - 09/2010
% Programa com grupos de neuroides (nrd) que formam memória binária, operando como
% Flip-Flops digitais. Associados, eles formam contadores binários ou acumuladores.
% Finalidade: analisar assembleias funcionando como memória temporária, atuando como
% FFs, e que podem realizar tarefas computacionais como contar, criar ramificações e
% desmantelá-las, gerando uma cadeia hierárquica causalmente conectada.
% nNo modo de operação pode-se escolher contador binário ou acumulador (ver simMode).
% Topologia: As assbs estão divididas em assembleias (kernels) de 'nK' neuroides,
% os lo.s (1,2...nK) são usados para estímulos externos (evento INPUT).
% Um grupo de kernels forma um 'pool' que executa a função de um FF-JK. Aqui usa-se
% quatro pools, interconectados, formando um contador binário de 4 bits.
% OBS: Monta um gráfico (fig) e mostra valores na tela de comando ao final, que se
% referem ao MAX e MIN de corrente (pA) injetado como ruído na rede.
% Uma função do programa é avaliar ESTABILIDADE perante ruído de corrente injetada.
% Para avaliar: modifique o parâmetro 'NoiseFactor' abaixo.
% -----

clear; clc; % limpa vars do Matlab e limpa tela...
simMode=1; % simulation mode: 1=BinaryCounter 0=Stacker
GlobalResetTime=0; % Global Reset time burst (=0 not simulated)
Nk=5; % Num of neurons per kernel (3...12 neurons)
pti=80; % Pulse Time Interval (ms) (for Input pulse)
NoiseFactor=0.05; % Fator de ruído introduzido ao circuito...
simTime=1600; % tempo total de simulação

% Initial synaptic weights/ kic & kip = kernel factors between kernels
if Nk<3; Nk=3; elseif Nk>20; Nk=20; end;
Np=4*Nk; Nn=4*Np+3*Nk; % Num of neurons in a pool / Num total of neurons
sw=70/Nk; % synaptic weight for all connections = f(Nk)
if simMode==1 % calcula um fator de acoplamento do peso sináptico
    kic=sw*0.071; % do kernel biestável p/ inter-neuroide (reset)
    kip=sw*0.135; % do (inter-kernel) IN para os nrds do reset
else
    kic=sw*0.071;
    kip=sw*0.081;
end;
MM=zeros(simTime,2);
% Parameters: Excitatory neurons are Regular Spiking (neuron model by Izhikevich)
a=0.01*ones(Nn,1); b=0.2*ones(Nn,1); c=-62*ones(Nn,1);
d=0.5*ones(Nn,1); v=-62*ones(Nn,1);
% inhibitoryneurons are Fast Spiking
for k=Np-Nk+1:Np
    a(k:Np:Nn,1)=0.1; b(k:Np:Nn,1)=0.2; c(k:Np:Nn,1)=-65;
    d(k:Np:Nn,1)=2; v(k:Np:Nn,1)=-65;
end;

% Initial values of u
u=b.*v;
% -- Synaptic weights (the network topology) -- %
S=zeros(Nn,Nn);
for j=0:3 % this sim has up to 4 pools
    i=j*Np+1; % index for ith neuron in the jth pool
    if simMode==1 % Binary Counting mode
        % Neural connections in the jth pool
        if j>=1; S(i:i+Nk-1,j*Np-Nk+1:j*Np)=sw; end;
        % Connections for the resonant memory cell neurons
        S(i+Nk:i+2*Nk-1,i:i+Nk-1)=sw; % jth kernel input signal
        S(i+3*Nk:i+4*Nk-1,i:i+Nk-1)=kip; % pool pulse input
        S(i+2*Nk:i+3*Nk-1,i+Nk:i+2*Nk-1)=sw; % biestable kernel_1 (jBk1)
        S(i+Nk:i+2*Nk-1,i+2*Nk:i+3*Nk-1)=sw; % biestable kernel_0 (jBk0)
        S(i+Nk:i+2*Nk-1,i+3*Nk:i+4*Nk-1)=-sw; % Internal reset for jBk0
        S(i+2*Nk:i+3*Nk-1,i+3*Nk:i+4*Nk-1)=-sw; % Internal reset for jBk1
        S(i+Nk:i+2*Nk-1,Nn-Nk+1:Nn)=-sw; % Global reset for jBk0
        S(i+2*Nk:i+3*Nk-1,Nn-Nk+1:Nn)=-sw; % Global reset for jBk1
        % Internal Inhibitory neurons for each pool
        S(i+3*Nk:i+4*Nk-1,i+Nk:i+3*Nk-1)=kic; % connection to jBk0 & jBk1
    else

```

```

% Neural connections in the jth pool
if j>=1; S(i:i+Nk-1,j*Np-3*Nk+1:j*Np-Nk)=kic; end;
% Connections for the resonant memory cell neurons
S(i+Nk:i+2*Nk-1,i:i+Nk-1)=sw; % jth kernel input signal
S(i+2*Nk:i+3*Nk-1,i+Nk:i+2*Nk-1)=sw; % biestable kernel_1 (jBk1)
S(i+Nk:i+2*Nk-1,i+2*Nk:i+3*Nk-1)=sw; % biestable kernel_0 (jBk0)
S(i+Nk:i+2*Nk-1,Nn-Nk+1:Nn)=-sw; % Global reset for jBk0
S(i+2*Nk:i+3*Nk-1,Nn-Nk+1:Nn)=-sw; % Global reset for jBk1
S(i:i+Nk-1,i+Nk:i+3*Nk-1)=-kip; % disable the kernel from IN signal
end;
end;
% Input signal for the first pool <- from the last neurons in the topology
if simMode==1
    S(1:Nk,Nn-2*Nk+1:Nn-Nk)=sw;
    S(3*Nk+1:4*Nk,Nn-3*Nk+1:Nn-2*Nk)=kip/10;
else
    S(1:Nk,Nn-2*Nk+1:Nn-Nk)=sw;
    S(1:Nk,Nk+1:3*Nk-1)=-sw; % disable the kernel from IN signal
    S(Np+1:Np+Nk,Nn-2*Nk+1:Nn-Nk)=kip;
    S(Np+1:Np+Nk,Nn-3*Nk+1:Nn-2*Nk)=kip;
    S(2*Np+1:2*Np+Nk,Nn-2*Nk+1:Nn-Nk)=kip;
    S(2*Np+1:2*Np+Nk,Nn-3*Nk+1:Nn-2*Nk)=kip;
    S(3*Np+1:3*Np+Nk,Nn-2*Nk+1:Nn-Nk)=kip;
    S(3*Np+1:3*Np+Nk,Nn-3*Nk+1:Nn-2*Nk)=kip;
end;

% ---- main simulation ---- %
Spks=[];
tt=4; % time for the first input pulse
for t=1:simTime % simulation tim in miliseconds
    I=NoiseFactor*randn(Nn,1); % small Input noises (pA)
    MM(t,1)=max(I);MM(t,2)=min(I); % para analisar min e max I (pA)
    % timing check for triggering/reset the neurons
    if t==tt
        v(Nn-2*Nk+1:Nn-Nk,1)=30; % this generate input pulse
    elseif t==tt+1
        tt=tt+pti;
        v(Nn-3*Nk+1:Nn-2*Nk,1)=30; % input pulse with delay 1ms
    elseif GlobalResetTime~=0 & (t>=GlobalResetTime&t<=GlobalResetTime+5)
        v(Nn-Nk+1:Nn,1)=30; % this forces the Global Reset burst
    else
        v(Nn-2*Nk-1:Nn-1,1)=-65; % this forces no input pulse
    end
    fired=find(v>=30); % index for the fired neuron(s)
    Spks=[Spks;t+0*fired,fired];
    v(fired,1)=c(fired,1); % reset of v() for fired neurons
    u(fired,1)=u(fired,1)+d(fired,1); % u() after neurons fired
    I=I+sum(S(:,fired),2); % In(mV) of each fired neuron
    v=v+0.5*(0.04*v.^2+5*v+140-u+I); % step 0.5 ms
    v=v+0.5*(0.04*v.^2+5*v+140-u+I); % for numerical stability
    u=u+a.*(b.*v-u); % u for the next interaction
end;
plot(Spks(:,1),Spks(:,2),'.');
['Max= ' num2str(max(MM(:,1))) ' / Min= ' num2str(min(MM(:,2))) ]

```

APÊNDICE C

Código MATLAB para Memória Biestável e Contadores COM DELAY

```
% MEMÓRIA BIESTÁVEL E CONTADORES BINÁRIOS COM DELAY: João Ranhel - V.4 - 05/2011
% Programa para estudar efeito do delay nas assembleias contadoras biestáveis.
% Finalidade: Demonstrar que os contadores biestáveis passam a funcionar como
% PCO (pulse coupled oscillators) quando se insere delays entre assembleias
% Topologia: As assbs estão divididas em assembleias (kernels) de 'nK' neuroides,
% os lo.s (1,2...nK) são usados para estímulos externos (evento INPUT).
% Um grupo de kernels forma um 'pool' que executa a função de um FF-JK. Aqui usa-se
% quatro pools, interconectados, formando um contador binário de 4 bits.
% ATENÇÃO: prog em LOOP, p/ interromper, no Command Window: press Ctrl+'Pause/Break'
% OBS 1: com DELAY o circuito NÃO FUNCIONA como contador binário, só memória biestável
% _2: CLIQUE sobre a figura (ou tecle) p/ próximo intervalo PULSO de ESTÍMULO.
% -----
clear; clc; % reiniciar Matlab

% -- 1. CONFIGURAÇÃO da simulação -- %
simMod=1; % modo de simulação: 1=BinaryCounter 0=StackerCounter
dIK=10; % atraso inter-kernel (inter-neuroids) delay
nK=5; % Num de nrds por kernel (3 <= nK <= 20 nrds)
fatorRuido=0.05; % Fator de ruído (random normalz) injetado na rede
% -- parametros temporais da simulação -- %
tsim=500; % tempo (ms) total de simulação
maxPD=20; % Tempo máx de atraso na propagação axonal dos spikes
globalResetTime=00; % Tempo (ms) para ocorrer reset global (=0 não simulado)
firstIN=50; % atraso para o 1o. INPUT (1o. pulso)
pINTi=104; % valor inicial p/ intervalo evento INPUT (será incrementado)
pINTf=140; % valor final p/ intervalo evento INPUT
% uma assembleia (kernel) é composta de nK nrds, um pool tem 4 kernels
if nK<3; nK=3; elseif nK>20; nK=20; end;
nP=4*nK; % cada pool contém 4 kernels
nN=4*nP+3*nK; % Num total de nrds na simulação
% relativo à distribuição de pesos sinápticos em relação ao num neuroids
sw0=68/nK; % sw p/ todas as sinapses = f(nK)
if simMod==1
    kic=sw0*0.071; % fator de acoplamento p/ reset interneuroides
    kip=sw0*0.135; % fator de acoplamento dos (inter-kernel) IN signal -> RESET
else
    kic=sw0*0.071;
    kip=sw0*0.081;
end;
% -- fim configuração -- %

% -- 2. Inicializa PARAMETROS dos neuroides -- %
% Matriz com os parâmetros para os 23 classes de nrds - Izhikevich, Simple Model
Pars=[0.02 0.2 -65 6 -70 14 ;... % 1 tonic spiking
0.02 0.25 -65 6 -64 0.5 ;... % 2 phasic spiking
0.02 0.2 -50 2 -70 15 ;... % 3 tonic bursting
0.02 0.25 -55 0.05 -64 0.6 ;... % 4 phasic bursting
0.02 0.2 -55 4 -70 10 ;... % 5 mixed mode
0.01 0.2 -65 8 -70 30 ;... % 6 spike freq adaptation
0.02 -0.1 -55 6 -60 0 ;... % 7 Class 1
0.2 0.26 -65 0 -64 0 ;... % 8 Class 2
0.02 0.2 -65 6 -70 7 ;... % 9 spike latency
0.05 0.26 -60 0 -62 0 ;... % 10 subthreshold oscillations
0.1 0.26 -60 -1 -62 0 ;... % 11 resonator
0.02 -0.1 -55 6 -60 0 ;... % 12 integrator
0.03 0.25 -60 4 -64 0 ;... % 13 rebound spike
0.03 0.25 -52 0 -64 0 ;... % 14 rebound burst
0.03 0.25 -60 4 -64 0 ;... % 15 threshold variability
1 1.5 -60 0 -61 -65 ;... % 16 bistability
1 0.2 -60 -21 -70 0 ;... % 17 DAP
0.02 1 -55 4 -65 0 ;... % 18 accommodation
-0.02 -1 -60 8 -63.8 80 ;... % 19 inh-induced spiking
-0.026 -1 -45 0 -63.8 80 ;... % 20 inh-induced bursting
0.01 0.2 -62 0.5 -62 10 ;... % 21 Regular Spiking
0.1 0.2 -65 2 -65 10 ;... % 22 Fast Spiking
0.02 -0.1 -55 6 -55 0] ; % 23 Class XG
% Parâmetros dos nrds EXCITATÓRIOS: 'Regular Spiking' ('Simple Model' de Izhikevich)
```

```

ipe=21; % ATENÇÃO: ESCOLHER um indx para o tipo de NEUROIDE a ser testado
A=Pars(ipe,1)*ones(nN,1); B=Pars(ipe,2)*ones(nN,1);
C=Pars(ipe,3)*ones(nN,1); D=Pars(ipe,4)*ones(nN,1);
% Parâmetros dos nrds INIBITÓRIOS: 'Fast Spiking'
ipi=22; % ATENÇÃO: ESCOLHER um indx para o tipo de NEUROIDE a ser testado
for k=nP-nK+1:nP
    A(k:nP:nN,1)=Pars(ipi,1); B(k:nP:nN,1)=Pars(ipi,2);
    C(k:nP:nN,1)=Pars(ipi,3); D(k:nP:nN,1)=Pars(ipi,4);
end;
% -- fim parametros dos neuroides -- %

% -- 3. TOPOLOGIA -- %
% e DA é a matriz com todos os delays entre nrds
DA=dIK*ones(nN,nN); % inicia todos (synfire chains)
% Inicia matrizes: Sw é a matriz com todos os pesos sinápticos
Sw=zeros(nN,nN);
for y=0:3 % simulação com 4 pools (4 bits counter)
    x=y*nP+1; % index p/ xth neuron na yth pool
    if simMod==1 % Binary Counting mode
        % conexões dos nrds entre os pools
        % do segundo pool para frente, faz conexão inter-pool
        if y>=1 Sw(y*nP-nK+1:y*nP,x:x+nK-1)=sw0; end
        Sw(x:x+nK-1,x+nK:x+2*nK-1)=sw0; % nrds IN -> kernel A
        Sw(x:x+nK-1,x+3*nK:x+4*nK-1)=kip; % nrds IN -> RESET
        Sw(x+nK:x+2*nK-1,x+2*nK:x+3*nK-1)=sw0; % nrds A -> B
        Sw(x+2*nK:x+3*nK-1,x+nK:x+2*nK-1)=sw0; % nrds B -> A
        % inibição interna
        Sw(x+3*nK:x+4*nK-1,x+nK:x+2*nK-1)=-sw0; % nrds RESET -> A
        Sw(x+3*nK:x+4*nK-1,x+2*nK:x+3*nK-1)=-sw0; % nrds RESET -> B
        % inibição global
        Sw(nN-nK+1:nN,x+nK:x+2*nK-1)=-sw0; % GlobalReset -> A
        Sw(nN-nK+1:nN,x+2*nK:x+3*nK-1)=-sw0; % GlobalReset -> B
        Sw(x+nK:x+3*nK-1,x+3*nK:x+4*nK-1)=kic; % A,B -> RESET
    else
        % Conexoes neurais entre pools
        % se y>1 interconectar com o pool anterior
        if y>=1 Sw(y*nP-3*nK+1:y*nP-nK,x:x+nK-1)=kic; end
        % conexões neurais intra pool
        Sw(x:x+nK-1,x+nK:x+2*nK-1)=sw0; % nrds IN -> kernel A
        Sw(x+nK:x+2*nK-1,x+2*nK:x+3*nK-1)=sw0; % nrds A -> B
        Sw(x+2*nK:x+3*nK-1,x+nK:x+2*nK-1)=sw0; % nrds B -> A
        Sw(nN-nK+1:nN,x+nK:x+2*nK-1)=-sw0; % GlobalReset -> A
        Sw(nN-nK+1:nN,x+2*nK:x+3*nK-1)=-sw0; % GlobalReset -> B
        Sw(x+nK:x+3*nK-1,x:x+nK-1)=-kip; % disable the kernel from IN signal
    end;
end;
% Input signal for the first pool <- from the last neurons in the topology
if simMod==1
    Sw(nN-2*nK+1:nN-nK,1:nK)=sw0; % INPUT_ev -> IN(A-1)
    Sw(nN-3*nK+1:nN-2*nK,3*nK+1:4*nK)=kip/10;
else
    Sw(nN-2*nK+1:nN-nK,1:nK)=sw0; % INPUT_ev -> IN(A-1)
    Sw(nK+1:3*nK-1,1:nK)=-sw0; % disable the kernel from IN signal
    Sw(nN-2*nK+1:nN-nK,nP+1:nP+nK)=kip;
    Sw(nN-3*nK+1:nN-2*nK,nP+1:nP+nK)=kip;
    Sw(nN-2*nK+1:nN-nK,2*nP+1:2*nP+nK)=kip;
    Sw(nN-3*nK+1:nN-2*nK,2*nP+1:2*nP+nK)=kip;
    Sw(nN-2*nK+1:nN-nK,3*nP+1:3*nP+nK)=kip;
    Sw(nN-3*nK+1:nN-2*nK,3*nP+1:3*nP+nK)=kip;
end;
% -- fim topologia -- %

% -- 4. SIMULAÇÃO -- %
for pINi=pINTi:pINTf % Variar INTERVALO do evento ESTÍMULO
    % Inicia matriz de recuperação de potencial de membrana
    V=Pars(ipe,5)*ones(nN,1); % reseta vars da membrana p/ nrds excit. e inib.
    for k=nP-nK+1:nP V(k:nP:nN,1)=Pars(ipi,5); end;
    U=B.*V;
    Spks=[]; % Ini matriz p/ salvar 'tempo' e 'nrd' disparados
    nxtINev=firstIN; % primeiro disparo do INPUT (estímulo entrada)
    for tsec=1:tsim % para 'tsec'=1 até o tempo total de simulação
        In=fatorRuido*randn(nN,1); % vetor CORRENTE sináptica (pA) (com ruído)
        % relativo a disparo do evento de INPUT
        if tsec==nxtINev
            V(nN-2*nK+1:nN-nK,1)=32; % faz V>30mv para nrd 'x' da assembleia INPUT
            %elseif tsec==nxtINev+1

```

```

    nxtINev=nxtINev+pINi; % ajusta para prox disparo
    %V(nN-3*nK+1:nN-2*nK,1)=32; % input pulse with delay lms
elseif globalResetTime~=0 & (tsec>=globalResetTime&tsec<=globalResetTime+5)
    V(nN-nK+1:nN,1)=32; % força uma rajada de RESET GLOBAL
else
    V(nN-2*nK-1:nN,1)=-65; % força pot membrana p/ NÃO ter INPUT
end

% Verificação se houve disparo de algum nrd
Fired=find(V>=30); % encontra fired (V>=30mv), colocar em Fired
Spks=[Spks;tsec+0*Fired,Fired]; % 'Spks' tem os spikes : guarda [tmp nrd]
V(Fired,1)=C(Fired,1); % reset membrana V() p/ todos nrds disparados
U(Fired,1)=U(Fired,1)+D(Fired,1); % re-calcula U() para os nrds disparados
% -- rotina que calcula quem vai disparar nesta iteração 'tsec'
ti=tsec-maxPD; if ti<1 ti=1; end % inicia ti ms atrás: (ti=tsec-maxPD)
te=tsec-1; % varrer disparos até lms atrás
for t=ti:te % para os últimos maxPD milissegundos
    Fired=Spks(find(Spks(:,1)==t),2); % ver nrds disparados em 't' (ti<=t<=(tt-1))
    if isempty(Fired) % se ninguém disparou no tempo 't',
        continue % apenas incrementa 't'
    end
    % ver se spike do nrd i atinge j agora, em 'tsec', considerando delays axonais
    for i=1:size(Fired,1) % para cada nrd (i) disparado em 't'
        % [num2str(tsec) ' / ' num2str(ti) ' / ' num2str(te) ' / ' num2str(t)]
        Rgi=t+1+DA(Fired(i,1),:); % vetor: tempo abs min entre i e cada j
        for j=1:nN % p/ todas as cols (nrds j são o alvo)
            if tsec==Rgi(1,j) % se spk i atinge j
                In(j,1)=In(j,1)+Sw(Fired(i,1),j); % acumula In+Sw() dos fired em 't'+ 'DA'
            end % if tsec
        end % for j
    end % for i
end % for 't'=ti:te
V=V+0.5*(0.04*V.^2+5*V+140-U+In); % step 0.5 ms (p/ estabilidade numérica)
V=V+0.5*(0.04*V.^2+5*V+140-U+In); % calc V para próxima iteração
U=U+A.*(B.*V-U); % U para a próxima iteração
end % for 'tsec'
% -- fim simulação -- %
plot(Spks(:,1),Spks(:,2),'.');
['Int: ' num2str(pINi) ' - ok']
z=waitforbuttonpress;
end

```


APÊNDICE D

Código MATLAB para estudos BÁSICOS de formação de assembleias com delay

```
% BÁSICO DE ASSEMBLEIAS: Joao Ranhel - V.1.4 - 14/10/2011
% Prog. com grupos de neuroides (nrd) interconectados formando assembleias (assb)
% sincronizadas (synfire) ou grupos policronizados (GP) (ver 'simMod').
% Finalidade: testar formação de assb ou coalizões, verificar fn PSEUDO-BOOLEANAS,
% testar ruído e falha de disparo na estabilidade das assembleias.
% As assbs são grupos de 'nNa' neuroides. ENTRE com 'numAssb'= num de assembleias
% Os lo.s neuroides (1...nNa) são usados para eventos de INPUT.
% O total de nrds é nN=(nNa*numAssb); todos interconectados com delays e Sw específicos
% Os últimos nrds (nN-nNa...nN) são usados para INIBIÇÃO (reset)
% As matrizes DA (delay axonal) e Sw (synaptic weight) contêm estrutura da rede.
% Topologia: DESENHE a topologia da rede por meio da matriz Topol[] da seguinte forma:
% Topol=[i1 j1 s1 d1 ip1; i2 j2 s2 d2 ip2; ... in jn sn dn ipn];
% onde: ix=assembleia pré, jx=assembleia pós, sx=peso sináptico entre ix->jx,
%       dx=delay axonal de ix->jx, e ipx=tipo de neuroide de jx (indx p/ tabela: Pars)
%       OBS: todas as conexões entre assembleias devem ser informadas!
% É possível simular synfire chain / grupos policronizados (ver simMod)
% -----
clear;
clc; % reinicia matlab vars e limpa tela

% -- 1. CONFIGURAÇÃO -- %
tsim=500; % Tempo da simulação (ms) para cada I aplicado ao nrd
maxPD=22; % max atraso de propagação (delay propagation) simulado
tBD=15; % tempo base para delay entre nrds (i) -> (j) (ms)
tbN=21; % TIPO de nrd (indx p/ Pars) (se preciso, mudar em Topol)
simMod=0; % modo: =0 sem atraso randômico (synfire chain)
%          =1 com atraso rand. (GPs policronizados)

% -- define num assembleias e nrds -- %
nNa=5; % Núm de neuroides/assembleia: 3<=nNa<=20
if nNa<3; nNa=3; elseif nNa>20; nNa=20; end; % restringe nNa
numAssb=5; % TOTAL de Assbs (cuidado p/ não usar indx > em Topol)
nN=nNa*numAssb;
sw0=68/nNa; % peso sináptico inicial p/ todas as conexões

% -- define tempos de estímulos (INPUT)-- %
tINev=40; % tempo no qual ocorre o primeiro evento de INPUT
tiINev=340; % intervalo entre eventos de INPUT (=0 não repete)
tGReset=252; % tempo para o reset global...
f_tINevRnd=0; % flag =1 adiciona aleatoriedade ao evento INPUT
TIEv=tINev*ones(nNa,1); % vetor c/ tempos de disparos dos nrds INPUT
if simMod==1 TIEv=4*randn(nNa,1)+TIEv(:,1); end
TIEv=uint32(TIEv);

% -- desenha TOPOLOGIA da rede %
% ex: '0'->'1'; '1'->'2'; '2'->'1' (loop); '1'->'3'; '2'->'3'; '4'->'1'; '4'->'2'.
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 1 sw0 tBD tbN;...
       1 3 sw0/2 1.5*tBD tbN; 2 3 sw0/2 0.5*tBD 0;...
       4 1 -sw0 tBD 0; 4 2 -sw0 tBD/2 0]; % estes últimos são global reset
% -- fim configuração -- %

% -- 2. ESTRUTURA DA REDE -- %
% Matriz com os parâmetros para os 23 classes de nrds - Simple Model: Izhikevich
Pars=[0.02 0.2 -65 6 -70 14 ;... % 1 tonic spiking
      0.02 0.25 -65 6 -64 0.5 ;... % 2 phasic spiking
      0.02 0.2 -50 2 -70 15 ;... % 3 tonic bursting
      0.02 0.25 -55 0.05 -64 0.6 ;... % 4 phasic bursting
      0.02 0.2 -55 4 -70 10 ;... % 5 mixed mode
      0.01 0.2 -65 8 -70 30 ;... % 6 spike freq adaptation
      0.02 -0.1 -55 6 -60 0 ;... % 7 Class 1
      0.2 0.26 -65 0 -64 0 ;... % 8 Class 2
      0.02 0.2 -65 6 -70 7 ;... % 9 spike latency
      0.05 0.26 -60 0 -62 0 ;... % 10 subthreshold oscillations
      0.1 0.26 -60 -1 -62 0 ;... % 11 resonator
      0.02 -0.1 -55 6 -60 0 ;... % 12 integrator
      0.03 0.25 -60 4 -64 0 ;... % 13 rebound spike
      0.03 0.25 -52 0 -64 0 ;... % 14 rebound burst
```

```

0.03      0.25    -60     4      -64     0    ;... % 15 threshold variability
1         1.5     -60     0      -61    -65    ;... % 16 bistability
1         0.2     -60    -21     -70     0    ;... % 17 DAP
0.02      1       -55     4      -65     0    ;... % 18 accommodation
-0.02     -1      -60     8      -63.8  80    ;... % 19 inh-induced spiking
-0.026    -1      -45     0      -63.8  80    ;... % 20 inh-induced bursting
0.01      0.2     -62     0.5    -62     10    ;... % 21 Regular Spiking
0.1       0.2     -65     2      -65     10    ;... % 22 Fast Spiking
0.02     -0.1     -55     6      -55     0]    ;    % 23 Class XG

A=Pars(tbN,1)*ones(nN,1); B=Pars(tbN,2)*ones(nN,1);
C=Pars(tbN,3)*ones(nN,1); D=Pars(tbN,4)*ones(nN,1); V=Pars(tbN,5)*ones(nN,1);

Sw=zeros(nN);           % Peso sináptico i -> j (i=pré (linha) e j=pós (coluna)
DA=int16(zeros(nN));     % Atraso na propagação do axônio (i) para a conexão pós (j)

% Assumiremos que os los. nrds [1,2,3...nN] fornecem os ESTÍMULOS!
for x=1:size(Topol,1)
    i=Topol(x,1); j=Topol(x,2);           % prox 2 assembleias i e j
    Sw(i*nNa+1:(i+1)*nNa,j*nNa+1:(j+1)*nNa)=Topol(x,3); % sw p/ conexoes assbs i e j
    if simMod==0
        DA(i*nNa+1:(i+1)*nNa,j*nNa+1:(j+1)*nNa)=Topol(x,4);
    else
        DA(i*nNa+1:(i+1)*nNa,j*nNa+1:(j+1)*nNa)=Topol(x,4)+4*randn;
    end
    ip=Topol(x,5);
    if ip>0&&ip<=23
        A(j*nNa+1:(j+1)*nNa,1)=Pars(ip,1);
        B(j*nNa+1:(j+1)*nNa,1)=Pars(ip,2);
        C(j*nNa+1:(j+1)*nNa,1)=Pars(ip,3);
        D(j*nNa+1:(j+1)*nNa,1)=Pars(ip,4);
        V(j*nNa+1:(j+1)*nNa,1)=Pars(ip,5);
    end
end
% -- fim estrutura -- %

In=zeros(nN,1);           % ini matriz das correntes sinápticas In dos nrd
U=B.*V;                   % Vals iniciais recuperação da V() da membrana

% -- 3. SIMULAÇÃO -- %
Spks=[];                   % Matriz p/ salvar 'tempo' e 'nrd' disparados
tsec=uint32(0);            % garante tsec=INT, resolução da simulação=1ms
for tsec=1:tsim            % para 'tsec'=1 até o tempo total de simulação
    if tsec>=tGReset&tsec<=tGReset+5 % se o Global Reset foi programado
        V(nN-nNa+1:nN,1)=32; % provoca uma rajada para reset
    else % procedimento que dispara eventos de INPUT (estímulos externos)
        if f_tINevRnd==0 nxtINev=uint32(0); % se flag p/ random no INPUT =0, nxtINev=0
        else nxtINev=uint32(2*randn); end % se flag=1, nxtINev desloca próx evento input
        TmpEvFire=find(TIEv==tsec); % teste: algum nrd do INPUT deve ser disparado?
        if ~isempty(TmpEvFire) % se tiver algum nrd p/ ser disparado
            V(TmpEvFire(:,1),1)=32; % faz V>30mv para nrd 'x' da assembleia INPUT
            TIEv(TmpEvFire(:,1),1)=TIEv(TmpEvFire(:,1),1)+tiINev+nxtINev;
        end
    end
    % Verificação se houve disparo de algum nrd
    Fired=find(V>=30); % encontrar disparados (V>=30mv), colocar em Fired
    Spks=[Spks; tsec+0:Fired,Fired]; % 'Spks' tem os spikes até 'tsec':guarda [tmp nrd]
    V(Fired,1)=C(Fired,1); % reseta membrana V() p/ todos os nrds disparados
    U(Fired,1)=U(Fired,1)+D(Fired,1); % re-calcula U() para os nrds disparados
    In=0.05*randn(nN,1); % vetor CORRENTE sináptica (pA) (com ruído)

    % -- rotina que calcula quem vai disparar nesta iteração 'tsec'
    ti=tsec-maxPD; if ti<1 ti=1; end % inicia ti ms atrás: (ti=tsec-maxPD)
    te=tsec-1; % varrer disparos até 1ms atrás
    for t=ti:te % para os últimos maxPD milissegundos
        Fired=Spks(find(Spks(:,1)==t),2); % ver nrds disparados em 't' (ti<=t<=(te-1))
        if isempty(Fired) % se ninguém disparou no tempo 't',
            continue % apenas incrementa 't'
        end
        % ver se spike do nrd i atinge j agora, em 'tsec', considerando delays axonais
        for i=1:size(Fired,1) % para cada nrd (i) disparado em 't'
            Rgi=uint32(t+1+DA(Fired(i,1),:)); % vetor: tempo abs min entre i e cada j
            for j=1:nN % p/ todas as cols (nrds j são o alvo)
                if tsec==Rgi(1,j) % se spk i atinge j
                    In(j,1)=In(j,1)+Sw(Fired(i,1),j); % acumula In+Sw() dos fired em 't'+DA'
                end % if tsec
            end
        end
    end
end

```

```

        end % for j
    end % for i
    end % for 't'=ti:te
    V=V+0.5*(0.04*V.^2+5*V+140-U+In); % passo 0.5 ms (p/ estabilidade numérica)
    V=V+0.5*(0.04*V.^2+5*V+140-U+In); % calc V para próxima iteração
    U=U+A.*(B.*V-U); % U para a próxima iteração
end % for 'tsec'
% -- fim SIMULAÇÃO -- %

plot(Spks(:,1),Spks(:,2),'.'); % raster plot da simulação!

['ok']

```

APÊNDICE E

Código MATLAB para assembleias com delay e grupos policronizados

```
% BÁSICO DE ASSEMBLEIAS: Joao Ranhel - V.1.5 - 25/10/2011
% Com grupos policronizados total (simMod=1) Escala de tempo diferente!
% -----
clear; clc; % reinicia matlab vars e limpa tela
% -- 1. CONFIGURAÇÃO -- %
tsim=800; % Tempo da simulação (ms) para cada I aplicado ao nrd
maxPD=50; % max atraso de propagação (delay propagation) simulado
tBD=44; % tempo base para delay entre nrds (i) -> (j) (ms)
tbN=21; % TIPO de nrd (indx p/ Pars) (se preciso, mudar em Topol)
simMod=1; % modo: =0 sem atraso randômico (synfire chain)
% =1 com atraso rand. (GPs policronizados)
nNa=4; % Núm de neuroides/assembleia: 3<=nNa<=20
if nNa<3; nNa=3; elseif nNa>20; nNa=20; end; % restringe nNa
numAssb=7; % TOTAL de Assbs (cuidado p/ não usar indx > em Topol)
nN=nNa*numAssb;
sw0=78/nNa; % peso sináptico inicial p/ todas as conexões
tINev=10; % tempo no qual ocorre o primeiro evento de INPUT
tiINev=int32(500); % intervalo entre eventos de INPUT (=0 não repete)
tGReset=0; % tempo para o reset global...
f_tINevRnd=0; % flag =1 adiciona aleatoriedade ao evento INPUT
fatorDispGP=2; % Fator de dispersão dos grupos policronais
TIEv=tINev*ones(nNa,1); % vetor c/ tempos de disparos dos nrds INPUT
if simMod==1 % se modo simulação é por grupos policronizados
    TmpTIEv=fatorDispGP*randn(nNa,1); % cria um vetor disparo randômico estímulo
    TIEv=TmpTIEv+TIEv; % monta o vetor do primeiro estímulo
    TmpTIEv=int32(TmpTIEv); % usa int32
end
TIEv=int32(TIEv); % usa int32
% -- desenha TOPOLOGIA da rede %
% ex: '0' -> '1'; '1' -> '2'; '2' -> '1' (loop); '1' -> '3'; '2' -> '3'; '4' -> '1'; '4' -> '2'.
Topol=[0 1 sw0 tBD tbN; 1 2 sw0 tBD tbN; 2 3 sw0 tBD tbN; 3 2 sw0 tBD tbN;
        2 4 sw0 tBD/2 tbN; 3 4 sw0 tBD/2 tbN;...
        2 5 sw0/4 tBD+tBD/4 tbN; 3 5 sw0/4 tBD/4 tbN;];
% OBS: [0 1 sw0 tBD tbN; => denota: assb '0' é conectada à '1' pelo peso sw0,
% e pelo tempo básico tBD, sendo que tbN é tipo de neuroide da assb '1'.
% -- 2. ESTRUTURA DA REDE -- %
% Matriz com os parâmetros para os 23 classes de nrds - Simple Model: Izhikevich
Pars=[0.02 0.2 -65 6 -70 14 ;... % 1 tonic spiking
      0.02 0.25 -65 6 -64 0.5 ;... % 2 phasic spiking
      0.02 0.2 -50 2 -70 15 ;... % 3 tonic bursting
      0.02 0.25 -55 0.05 -64 0.6 ;... % 4 phasic bursting
      0.02 0.2 -55 4 -70 10 ;... % 5 mixed mode
      0.01 0.2 -65 8 -70 30 ;... % 6 spike freq adaptation
      0.02 -0.1 -55 6 -60 0 ;... % 7 Class 1
      0.2 0.26 -65 0 -64 0 ;... % 8 Class 2
      0.02 0.2 -65 6 -70 7 ;... % 9 spike latency
      0.05 0.26 -60 0 -62 0 ;... % 10 subthreshold oscillations
      0.1 0.26 -60 -1 -62 0 ;... % 11 resonator
      0.02 -0.1 -55 6 -60 0 ;... % 12 integrator
      0.03 0.25 -60 4 -64 0 ;... % 13 rebound spike
      0.03 0.25 -52 0 -64 0 ;... % 14 rebound burst
      0.03 0.25 -60 4 -64 0 ;... % 15 threshold variability
      1 1.5 -60 0 -61 -65 ;... % 16 bistability
      1 0.2 -60 -21 -70 0 ;... % 17 DAP
      0.02 1 -55 4 -65 0 ;... % 18 accommodation
      -0.02 -1 -60 8 -63.8 80 ;... % 19 inh-induced spiking
      -0.026 -1 -45 0 -63.8 80 ;... % 20 inh-induced bursting
      0.01 0.2 -62 0.5 -62 10 ;... % 21 Regular Spiking
      0.1 0.2 -65 2 -65 10 ;... % 22 Fast Spiking
      0.02 -0.1 -55 6 -55 0 ;... % 23 Class XG
A=Pars(tbN,1)*ones(nN,1); B=Pars(tbN,2)*ones(nN,1);
C=Pars(tbN,3)*ones(nN,1); D=Pars(tbN,4)*ones(nN,1); V=Pars(tbN,5)*ones(nN,1);
Sw=zeros(nN); % Peso sináptico i -> j (i=pré (linha) e j=pós (coluna))
DA=zeros(nN); % Atraso na propagação do axônio (i) para a conexão pós (j)
% Assumiremos que os los. nrds [1,2,3...nNa] fornecem os ESTÍMULOS!
for x=1:size(Topol,1)
    i=Topol(x,1); j=Topol(x,2); % prox 2 assembleias i e j
```

```

Sw(i*nNa+1:(i+1)*nNa,j*nNa+1:(j+1)*nNa)=Topol(x,3); % sw p/ conexoes assbs i e j
if simMod==0
    DA(i*nNa+1:(i+1)*nNa,j*nNa+1:(j+1)*nNa)=Topol(x,4);
else
    % se simMod==1 então para cada nrd tem que forçar uma relação temporal
    TmpIEv=int32(fatorDispGP*randn(nNa,1));
    y3=0;
    for y1=(i*nNa+1):(i+1)*nNa
        y3=y3+1;
        y4=0;
        for y2=j*nNa+1:(j+1)*nNa
            y4=y4+1;
            DA(y1,y2)=Topol(x,4)-TmpTIEv(y3,1)+TmpIEv(y4,1);
        end
    end
end
ip=Topol(x,5);
if ip>0&&ip<=23
    A(j*nNa+1:(j+1)*nNa,1)=Pars(ip,1);    B(j*nNa+1:(j+1)*nNa,1)=Pars(ip,2);
    C(j*nNa+1:(j+1)*nNa,1)=Pars(ip,3);    D(j*nNa+1:(j+1)*nNa,1)=Pars(ip,4);
    V(j*nNa+1:(j+1)*nNa,1)=Pars(ip,5);
end
end
In=zeros(nN,1); % ini matriz das correntes sinápticas In dos nrd
U=B.*V; % Vals iniciais recuperação da V() da membrana
% -- 3. SIMULAÇÃO -- %
Spks=[]; % Matriz p/ salvar 'tempo' e 'nrd' disparados
tci=uint32(0); % garante tci=INT, resolução da simulação=lms
for tci=1:tsim % para 'tci'=1 até o tempo total de simulação
    if tci>=tGReset-2&tci<=tGReset+2 % se o Global Reset foi programado
        V(nN-nNa+1:nN,1)=32; % provoca uma rajada para reset
    else % procedimento que dispara eventos de INPUT (estímulos externos)
        if f_tINevRnd==0 nxtINev=int32(0); % se flag p/ random no INPUT =0, nxtINev=0
        else nxtINev=int32(2*randn); end % se flag=1, nxtINev desloca próx evento input
        TmpEvFire=int32(find(TIEv==tci)); % teste: algum nrd do INPUT deve ser disparado?
        if ~isempty(TmpEvFire) % se tiver algum nrd p/ ser disparado
            V(TmpEvFire(:,1),1)=32; % faz V>30mv para nrd 'x' da assembleia INPUT
            TIEv(TmpEvFire(:,1),1)=TIEv(TmpEvFire(:,1),1)+tiINev+nxtINev;
        end
    end
    % Verificação se houve disparo de algum nrd
    Fired=find(V>=30); % encontrar disparados (V>=30mv), colocar em Fired
    Spks=[Spks; tci+0:Fired,Fired]; % 'Spks' tem os spikes até 'tci':guarda [tmp nrd]
    V(Fired,1)=C(Fired,1); % reseta membrana V() p/ todos os nrds disparados
    U(Fired,1)=U(Fired,1)+D(Fired,1); % re-calcula U() para os nrds disparados
    In=0.05*randn(nN,1); % vetor CORRENTE sináptica (pA) (com ruído)
    % -- rotina que calcula quem vai disparar nesta iteração 'tci'
    ti=tci-maxPD; if ti<1 ti=1; end % inicia ti ms atrás: (ti=tci-maxPD)
    te=tci-1; % varrer disparos até lms atrás
    for t=ti:te % para os últimos maxPD milissegundos
        Fired=Spks(find(Spks(:,1)==t),2); % ver nrds disparados em 't' (ti<=t<=(tt-1))
        if isempty(Fired) % se ninguém disparou no tempo 't',
            continue % apenas incrementa 't'
        end
        % ver se spike do nrd i atinge j agora, em 'tci', considerando delays axonais
        for i=1:size(Fired,1) % para cada nrd (i) disparado em 't'
            Rgi=uint32(t+1+DA(Fired(i,1),:)); % vetor: tempo abs min entre i e cada j
            for j=1:nN % p/ todas as cols (nrds j são o alvo)
                if tci==Rgi(1,j) % se spk i atinge j
                    In(j,1)=In(j,1)+Sw(Fired(i,1),j); % acumula In+Sw() dos fired em 't'+DA'
                end % if tci
            end % for j
        end % for i
    end % for 't'=ti:te
    V=V+0.5*(0.04*V.^2+5*V+140-U+In); % passo 0.5 ms (p/ estabilidade numérica)
    V=V+0.5*(0.04*V.^2+5*V+140-U+In); % calc V para próxima iteração
    U=U+A.*(B.*V-U); % U para a próxima iteração
end % for 'tci'
% -- fim SIMULAÇÃO -- %
plot(Spks(:,1),Spks(:,2),'.'); % raster plot da simulação!

```

Para versão atualizada (inglês): http://www.neuralassembly.org/download/NAC_Fundamentals.m

Tutorial (inglês): http://www.neuralassembly.org/download/Tutorial_NAC_Fundamentals.pdf

APÊNDICE F: http://www.neuralassembly.org/download/IEEE_TSMC_CwSNA.pdf