Bio-neurale netwerken

Pronk, M.

Bio-neurale Netwerken

Matthijs Pronk

6 oktober 2006

Inhoudsopgave

1 Inleiding

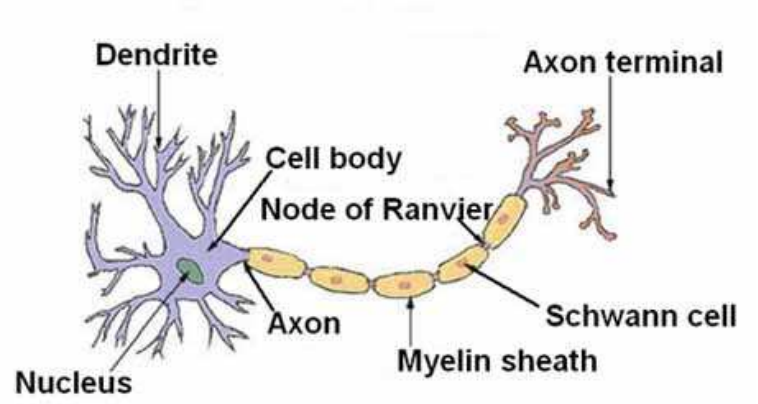
In de wat meer gecompliceerde organismen, zoals de mens, vormen netwerken van zenuwcellen een signaalsysteem, voor het observeren en reageren op de buitenwereld. Zo'n netwerk van zenuwcellen noemen we een bio-neuraal netwerk. We bekijken zenuwcellen als lichaamscellen die een impuls kunnen ontvangen, en die een impuls kunnen doorgeven aan een andere zenuwcel, waarmee die in contact staat. We kijken in deze scriptie naar het potentiaalverschil van een netwerk van zenuwcellen, uitgezet tegen de tijd.

Om inzicht te krijgen in zulk soort modellen, moeten we ons eerst beperken tot de werking van een enkele zenuwcel. Hierbij zullen we numerieke experimenten uitvoeren met CONTENT. Vervolgens zullen we uitgebreid een wiskundig model voor de membraanpotentialen van de cellen in een bio-neuraal netwerk afleiden, en ook daarmee numerieke experimenten uitvoeren. We zullen ons bezighouden met twee versies van het model: een versie zonder tijdsvertraging, delay, en een versie met delay. De delay ontstaat uit het feit dat het tijd kost om als prikkel van het ene neuron naar het andere te reizen. Het doel van het onderzoek is, behalve het 'verkennen' van deze modellen, het uitzoeken van belangrijke verschillen tussen de twee modellen. De numerieke analyse van het tweede model, het model waar er sprake is van een vertraging, zal worden gedaan met Matlab. Hierbij heb ik zelf een programma binnen Matlab geschreven, die een delay-vergelijking kan oplossen met behulp van een methode die Successieve Integratie heet. Dit bespreken we, en in de appendix zit de code van de m-file.

2 Het enkele neuron

We bekijken in dit hoofdstuk de bouw en werking van een enkele neuron. Uit dat resultaat halen we een differentiaal vergelijking voor het potentiaalverschil over de celmembraan van het cellichaam, zodat we een beeld krijgen van de vorm van prikkels die de cel uit gaan.

2.1 De bouw

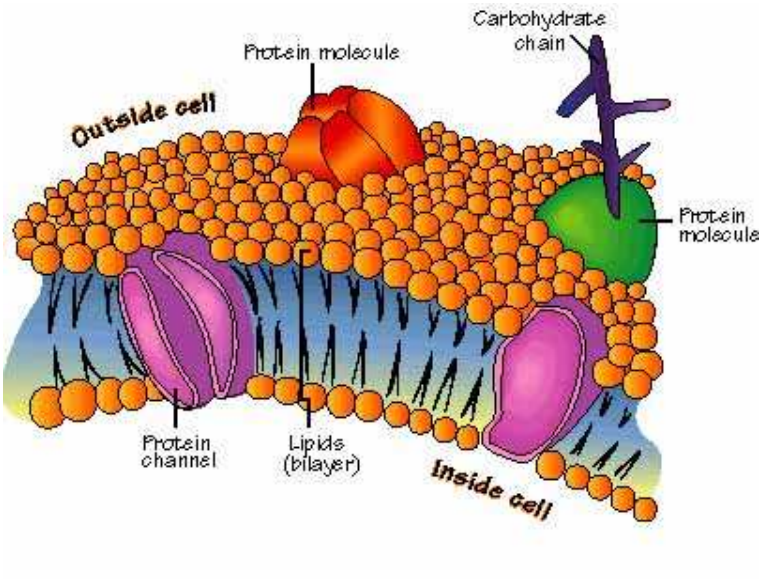


Figuur 1: de bouw van een neuron

Links op de figuur (Figuur 1) is het cellichaam met daarin de celkern te zien. De vertakkingen links hiervan noemen we de dendrieten. De prikkel die aankomt in deze cel, komt aan bij deze dendrieten in de vorm van een verhoogde of verlaagde potentiaal. Rechts van het cellichaam is het axon te zien, dit is vaak een lange uitloper. De overgang van cellichaam naar axon noemen we het axon hillock. Grenzend aan het axon zijn andere vertakkingen die we synapsen noemen. Deze synapsen staan in contact met de dendriet van een andere neuron.

2.2 De werking

De werking van een neuron begint bij de dendrieten. We moeten eerst even inzoomen op de celwand. Zie Figuur 2. De celwand bestaat uit een dubbele vetlaag. Zo'n vet bestaat uit een glycerol-molecuul met daaraan drie vetzuren. Het glycerol-molecuul is hydrofiel, dat wil zeggen water-aantrekkend. De vetzuren zijn hydrofoob, dat wil zeggen water-afstotend. De dubbele vetlaag zorgt er hier voor dat het glycerol-molecuul aan de beide buitenkanten van de celwand zit, omdat het cellichaam voor een groot deel uit water bestaat. De glycerol-moleculen zijn op de figuur duidelijk te zien als oranje bolletjes. Behalve vet-moleculen bestaat de celwand ook uit eiwitkanalen. Deze kanalen zorgen voor een verkeersstroom aan ionen zoals Ca^{2+}, K^+, Na^+, Cl^-. Deze ionenstroom



Figuur 2: de bouw van een neuron

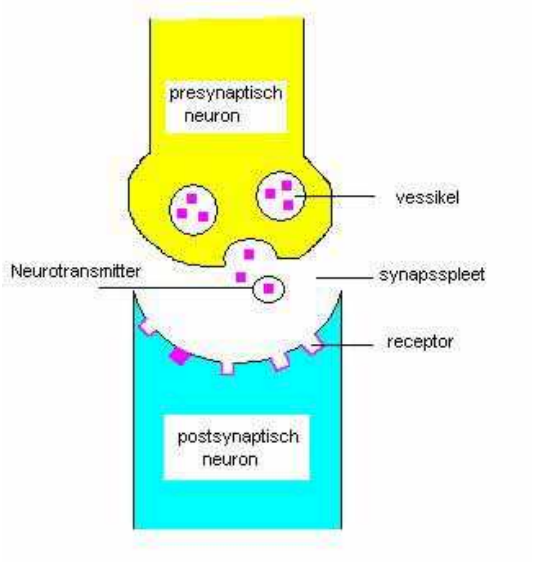
zorgt voor een potentiaalverschil tussen binnen- en buiten de cel. Dit verschil bedraagt ongeveer $-65mV$. Dit potentiaal noemen we de *rustpotentiaal*.

Als het neuron op een goede manier wordt geprikkeld (wat dat betekent wordt hieronder uitgelegd), zal er een zogenaamde *potentiaalpiek* ontstaan, een spike, die over het cellichaam loopt, in de richting van het axon. Bij het axon hillock wordt deze spike omgezet in een *pulstrein*. Deze loopt over het axon in de richting van de synapsen. In de synaps zitten kleine zakjes met daarin *neurotransmitters*. Deze zakjes noemen we *vesicles*. Onder invloed van de pulstrein bewegen de vesicles in de richting van de celmembraan. Daar aangekomen worden de neurotransmitters losgelaten in de intercellulaire omgeving. De neurotransmitters komen aan bij de dendrieten van het volgende neuron. En zo wordt de puls aan de volgende cel doorgegeven.

2.3 Twee soorten neurotransmitters

Er zijn grofweg twee soorten neurotransmitters: één soort dat de activiteit van een neuron stimuleert, en één soort dat de activiteit van een neuron remt. In beide gevallen zorgen ze voor een verandering in de rustpotentiaal. De stimulerende transmitters zorgen voor een stijging in het rustpotentiaal terwijl de remmende transmitters zorgen voor een daling.

In ieder neuron is er een bepaalde drempelwaarde, die bereikt moet worden om een spike, ofwel actie-potentiaal te activeren. Dit gaat volgens een



Figuur 3: de bouw van een neuron

'alles-of-niets' principe: wordt de drempelwaarde bereikt, dan zal er altijd een zelfde soort spike ontstaan, wordt de drempelwaarde niet bereikt, dan gebeurt er verder ook niets. Dat is belangrijk, want we willen dat straks in ons model terug kunnen zien.

2.4 Het afleiden van de differentiaalvergelijking

Het doel van de scriptie is niet het kijken naar een enkele neuron, daarom zullen we niet al te diep ingaan op het afleiden van deze differentiaalvergelijking. Voor details, zie [1]. Een groot deel van de vergelijking, onder meer de parameterwaarden, is verkregen via experimenten.

Alhoewel er meerdere ionen bijdrage aan de ionenstroom die de rustpotentiaal veroorzaakt, wordt dit in de differentiaalvergelijking gemodelleerd via een enkele *leak conductance* g_L door de celmembraan. Behalve deze passieve ionenstroom krijgen we ook te maken met de actieve ionenstroom van ionen Na^+ en K^+. Deze actieve ionenstroom hangt onder meer af van de eiwitkanalen. Deze kanalen kunnen namelijk open en dicht gaan. We introduceren functies $\alpha(V)$ als de mate waarin een open kanaal zal sluiten, en $\beta(V)$ als de mate waarin een gesloten kanaal zal openen. Experimenten suggereren dat er twee soorten deeltjes zijn die het kanaal kunnen openen.

We komen uiteindelijk op het volgende gekoppelde vierdimensionale stelsel

differentiaalvergelijkingen, de Hodgkin-Huxley vergelijkingen:

$$\begin{aligned}
C\frac{dV}{dt} &= g_L(V_L - V) + \bar{g}_{Na}m^3h(V_{Na} - V) + \bar{g}_K n^4(V_K - V) \\
\frac{dm}{dt} &= \alpha_m(V)(1-m) - \beta_m(V)m \\
\frac{dh}{dt} &= \alpha_h(V)(1-h) - \beta_h(V)h \\
\frac{dn}{dt} &= \alpha_n(V)(1-n) - \beta_n(V)n
\end{aligned}$$

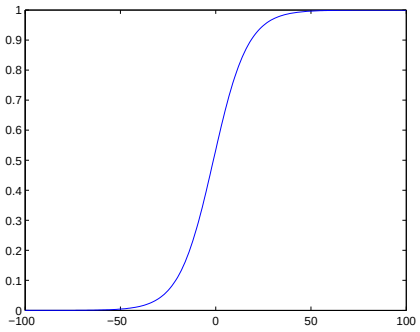
Echter omdat dit een vierdimensionale vergelijking is, is het lastiger om numerieke analyse te doen. Daarom nemen we een ander model, die gebaseerd is op een calciumkanaal in plaats van een natriumkanaal. Het calciumkanaal kan als instantaan worden beschouwd, en daarmee verkrijgen we een tweedimensionaal systeem, het Morris-Lecar model:

$$C\frac{dV}{dt} = I + g_L(V_L - V) + g_{Ca}m_\infty(V)(V_{Ca} - V) + g_K w(V_K - V) \tag{1}$$

$$\frac{dw}{dt} = \frac{\phi(w_\infty(V) - w)}{\tau(V)} \tag{2}$$

Hierbij is $m_\infty(V)$ de fractie van open calciumkanalen. Dit is een functie van V en niet van de tijd. w is de fractie van open kaliumkanalen. Vanuit [2] hebben we de functies

$$\begin{aligned}
m_\infty &= 0.5[1 + \tanh((V - v_1)/v_2)], \\
w_\infty &= 0.5[1 + \tanh((V - v_3)/v_4)], \\
\tau &= \frac{1}{\cosh((V - v_3)/(2v_4))}.
\end{aligned}$$



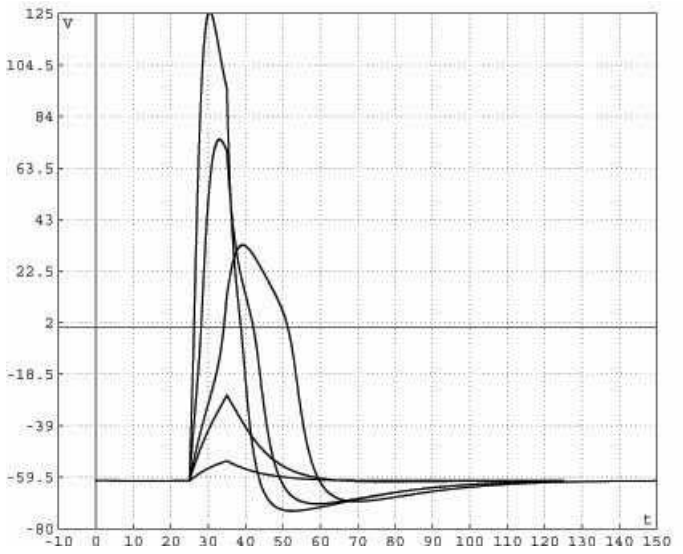
Figuur 4: de fractie open kanalen tegen de potentiaal V, parameterwaarden $v_1 = -1.2, v_2 = 18$.

3 Fasevlak Analyse van het enkele neuron

Met behulp van het stelsel (1), (2) kunnen we numerieke analyse uitvoeren. De parameterwaarden hebben we verkregen uit [2]. Zie ook de tabel hieronder:

Parameter	**Value**
C	$20\mu F/cm^2$
V_K	$-84mV$
g_K	$8mS/cm^2$
V_{Ca}	$120mV$
g_{Ca}	$4.4mS/cm^2$
V_{leak}	$-60mV$
g_{leak}	$2mS/cm^2$
$v1$	$-1.2mV$
$v2$	$18mV$
$v3$	$2mV$
$v4$	$30mV$
ϕ	$0.04/ms$

Merk op dat de I uit (1) de stroom is, die van buitenaf wordt toegevoegd. Uit de theorie weten we dat, zonder toevoegen van pulsen van buitenaf, na verloop van tijd de rustpotentiaal moet worden aangenomen. (De hoeveelheid tijd die daarvoor nodig is hangt natuurlijk af van de beginwaarde $V(0) = V_0$)
Met CONTENT meten we de waarde $V_{rest} = -60.558$. We zetten deze rustpotentiaal als beginwaarde: $V_0 = V_{rest}$. Zo kunnen we gaan analyseren, hoeveel I we moeten toevoegen om bij het enkele neuron een piek te doen ontstaan. In Figuur 5 zien we 5 verschillende oplossingen, bij 5 verschillende waarden van I. Eerst lieten we de oplossing tot $t = 25$ zonder puls van buitenaf, oftewel met $I = 0$. Van $t = 25$ tot $t = 35$ hebben we een positieve waarde van I ingevoerd en de oplossing in dat interval bekeken, en vanaf $t = 35$ zetten we weer $I = 0$.



Figuur 5: 5 verschillende oplossingen, bij 5 verschillende waarden van I

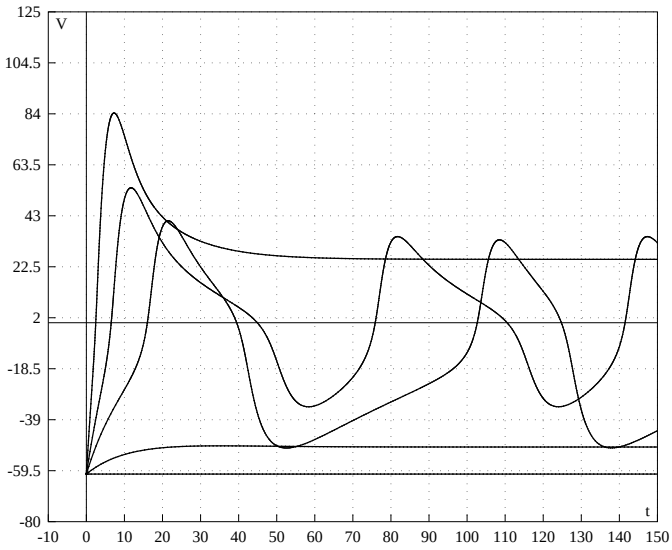
Er geldt: hoe hoger de piek komt, hoe groter de waarde van I. We hebben gebruikt: $I = 25, I = 100, I = 150, I = 400, I = 1000$. I wordt gemeten in pA. (De onderste curve hoort dus bij $I = 25$ terwijl de bovenste bij $I = 1000$ hoort.)

Het volgende dient uit de figuur te worden opgemerkt: bij de 'middelste' curve, dat is de curve met $I = 150$ op het tijdsinterval $[25, 35]$, zien we dat de top van de grafiek niet in dit interval ligt, maar rechts van dit interval. Dat wil zeggen, volgens de theorie, dat in dit geval de drempelwaarde is overschreden. Het gevolg hiervan is dat het neuron een echte piek toont, terwijl dat bij de twee gevallen met lagere I niet zo is.

Hierboven hebben we de positieve I op een klein tijdsinterval 'aangezet' en buiten dat interval I op 0 gehouden. De volgende figuur, figuur 6, laat zien wat er gebeurt als we de positieve I aanhouden.

We zien in deze figuur (figuur 6) 3 stationaire oplossingen, waarvan er twee in het onderste deel van de diagram te zien zijn, en één in het bovenste gedeelte. Daartussen zien we 2 oscillerende oplossingen.
We zijn vooral geïnteresseerd in de hoeveelheid I die nodig is om een oplossing te laten oscilleren. Voor de evenwichtswaarde betekent dit, dat deze van stabiliteit moet veranderen. We gebruiken daarom bifurcatie-analyse en zoeken numeriek naar de Hopf-bifurcatie. Dat is uitstekend te doen met het softwarepakket CONTENT [1]. Zie de figuur. CONTENT geeft ons dat de eerste Hopf-bifurcatie plaatsvindt bij $I = 93.8576$ Verticaal is de potentiaal van het

[1] CONTENT is te vinden op: http://www.math.uu.nl/people/kuznet/CONTENT/

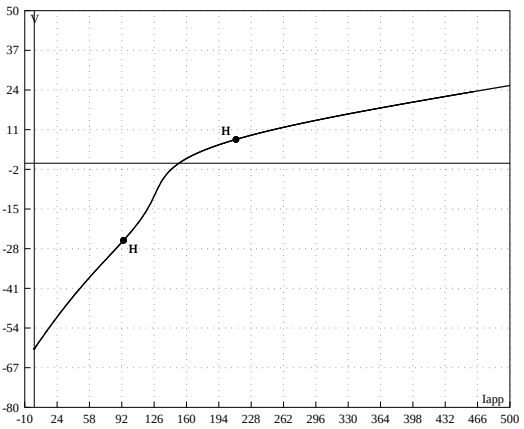


Figuur 6: Stationaire en oscillerende oplossingen

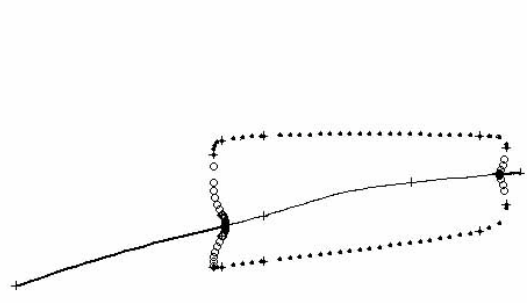
evenwicht weergegeven. Tussen de twee bifurcatiepunten in Figuur 7 is dit evenwicht instabiel.

Volgens de theorie zou er tussen de eerste Hopfbifurcatie bij $I = 93.8576$ en de tweede Hopf-bifurcatie bij $I = 212.019$ een stabiele limit cycle aanwezig moeten zijn. Er is echter meer. Voor beginwaarde $V = 90$ is er sprake van een stabiel evenwicht (Zie Figuur 7) maar ook een stabiele en een onstabiele limit cycle. De limit cycles zijn te zien in Figuur . De grootste limit cycle is stabiel terwijl de binnenste limit cycle onstabiel is. Aangekomen bij $I = 93.8576$ zal de binnenste limit cycle ingekrompen zijn tot een onstabiel evenwichtspunt, terwijl de buitenste stabiele limit cycle blijft bestaan.

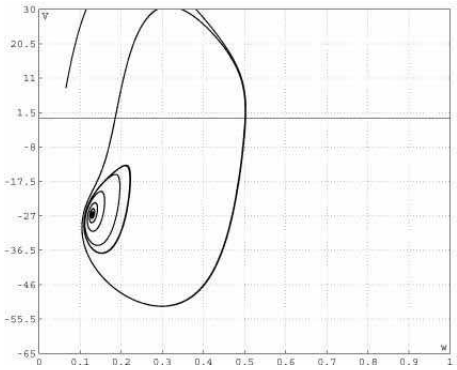
Ook voor $I > 212.019$ niet al te groot hebben we te maken met twee limit cycles. Voor nog grotere I verdwijnen beide limit cycles.
We merken op dat de vorm van de puls, die we in Figuur 5 zien, van belang is voor ons model van een bioneuraal netwerk.



Figuur 7: Hopf-bifurcatie met parameter I



Figuur 8: Hopf-bifurcatie samen met de limit cycle



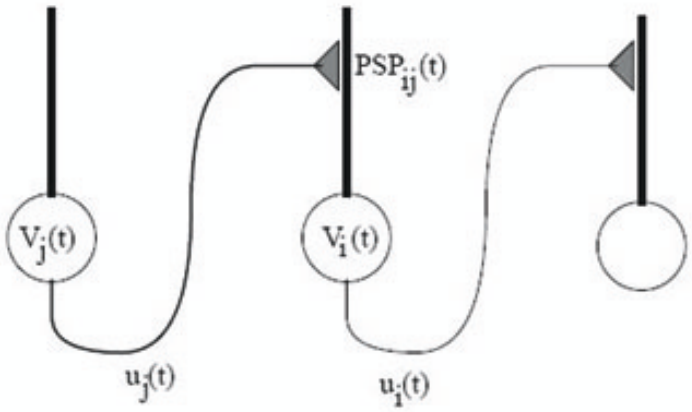
Figuur 9: Stabiele en onstabiele limit cycle, en een stabiel evenwichtspunt. $I = 90$

4 Een netwerk van neuronen

Aan de hand van de informatie die we uit het vorige hoofdstuk hebben verkregen, willen we het een en ander te weten komen over een netwerk van neuronen. We beginnen met het afleiden van de (gekoppelde) differentiaal vergelijking voor een netwerk van neuronen. Dit hoofdstuk is gebaseerd op [4].

4.1 De differentiaalvergelijking voor een netwerk van neuronen

Aan de hand van de kennis die we hebben omtrent een netwerk van neuronen en de differentiaalvergelijking voor één neuron, willen we een model maken voor een netwerk van neuronen. Om een model te maken dat nog redelijk te analyseren valt, moeten we een aantal vereenvoudigingen maken. Om het overzicht te behouden zetten wij hieronder een schematische weergave van een netwerk van neuronen (Figuur 10).

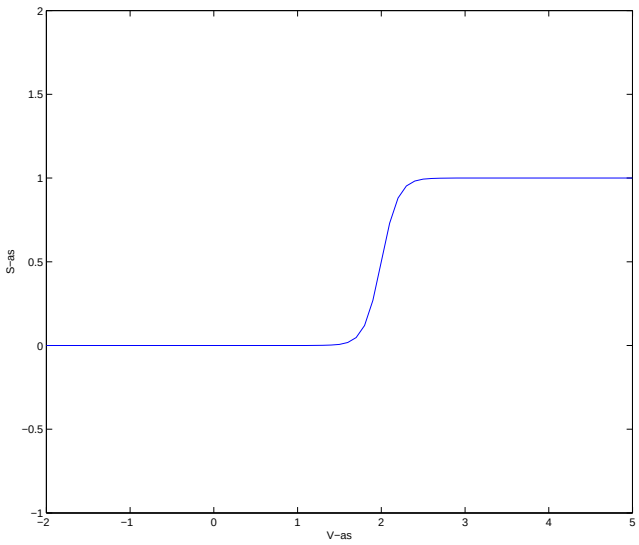


Figuur 10: Netwerk van neuronen schematisch

Herinner het volgende: bij prikkeling van neuron j ontstaat er op het axon (van neuron j) een pulstrein richting neuron i. Eenmaal aangekomen bij de synapsen en uiteindelijk bij de dendrieten van neuron i ontstaat er een potentiaalverandering op de ontvangende cel, de zogenaamde *Post Synaptic Potential*, afgekort PSP. Het aantal pulsen dat per tijdseenheid neuron j verlaat, is gegeven door de functie $u_j(t)$. Dit is dus een frequentie. Herinner, dat de omzetting van een puls op het neuron, V_j, in een pulstrein met frequentie u_j plaatsvindt in de 'axon hillock'. Deze omzetting modelleren we door een functionele afhankelijkheid: $u_j(t) = S_j(V_j(t))$.

Voor de functie S_j nemen we:

$$S_j(V_j) = \frac{S_{max}}{1 + e^{-\frac{V_j - V_S}{V_T}}}.$$



Figuur 11: Functionele vorm van S met $S_{max} = 1, V_T = 2, V_S = 0.1$

Zoals in de figuur (Figuur 11) te zien, stijgt de functie S van 0 naar S_{max}. Er is een omslagpunt, V_S, en hoe steil de functie daar loopt wordt gegeven door V_T. Merk op dat we veronderstellen dat de functionele vorm van S_j onafhankelijk is van j, i.e. deze is voor iedere neuron essentieel hetzelfde. We schrijven daarom S in plaats van S_j in het vervolg.

We veronderstellen dat iedere puls van neuron j eenzelfde postsynaptische potentiaal op neuron i bewerkstelligt, gegeven door de functie $t \mapsto PSP_{ij}(t)$. We veronderstellen verder dat de postsynaptic potentials van opeenvolgende pulsen simpelweg sommeren (superpositiehypothese). Als we er vanuit gaan dat op tijdstippen $t_1, t_2, \ldots$ pieken aankomen, dan hebben we dat de totale potentiaal wordt gegeven door

$$\Phi_{ij}(t) = \sum_k PSP_{ij}(t - t_k).$$

Het aantal pulsen dat op een interval $[t, t + dt]$ aankomt, wordt gegeven door $u_j(t)dt$. Hieruit volgt dat de potentiaal wordt gegeven door:

$$\Phi_{ij}(t) = \int_{t_0}^{t} PSP_{ij}(t - s)u_j(s - r_{ij})ds.$$

Hierbij is r_{ij} de vertraging die ontstaat, wanneer een puls over het axon moet "reizen".

Aangezien meerdere neuronen gekoppeld kunnen zijn aan neuron i, hebben we de hoofdvergelijking:

$$V_i(t) = \sum_j \int_{t_0}^{t} PSP_{ij}(t-s) S_j(V_j(s-r_{ij}))ds. \tag{3}$$

Omdat $u_j(t) = S(V_j(t))$, geldt er ook wel

$$u_i(t) = S_i \left(\sum_j \int_{t_0}^{t} PSP_{ij}(t-s) u_j(s-r_{ij})ds \right). \tag{4}$$

De vorm van de functie $PSP_{ij}(t)$ is die van een puls van een enkel neuron, wanneer deze geactiveerd wordt. Uit hoofdstuk 3 weten we hoe deze eruit ziet in het Morris-Lecar model ([1]). We maken vereenvoudigende aannamen zodat we uit de hoofdvergelijking een differentiaalvergelijking of delayvergelijking kunnen afleiden. Daartoe komen we op de volgende functionele vorm:

$$PSP_{ij}(t) = \begin{cases} w_{ij} e^{-\frac{t}{\tau_i}} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

De term w_{ij} bepaalt de sterkte van de puls, en of deze inhiberend dan wel exciterend is. We schrijven

$$PSP_{ij}(t) = w_{ij} E_i(t), \qquad E_i(t) = \begin{cases} e^{-\frac{t}{\tau_i}} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

In de volgende paragraaf gebruiken we dit om uit de hoofdvergelijking (3) een stelsel van differentiaalvergelijkingen of delayvergelijkingen af te leiden.

4.2 De operator K

We definiëren de integraaloperator

$$Ku(t) = \int_{t_0}^{\infty} E_i(t-s) u(s-r_{ij})ds,$$

en we bewijzen dat:

$$(\frac{d}{dt} + \frac{1}{\tau_i}) Ku(t) = u(t-r_{ij}).$$

Merk allereerst op dat

$$\int_{t_0}^{\infty} E_i(t-s) u(s-r_{ij})ds = \int_{t_0}^{t} E_i(t-s) u(s-r_{ij})ds$$

omdat voor waarden $s > t$ geldt dat $s - t < 0$. Hieruit volgt dat $E_i(t-s) = 0$.

4.2.1 Geval $h > 0$

Laat $h > 0$, er geldt:

$$\begin{aligned}\frac{Ku(t+h) - Ku(t)}{h} &= \int_{t_0}^{\infty} \frac{E_i(t+h-s) - E_i(t-s)}{h} u(s-r_{ij})ds \\ &= \int_{t_0}^{t+h} \frac{E_i(t+h-s) - E_i(t-s)}{h} u(s-r_{ij})ds \\ &= \int_{t_0}^{t} \frac{E_i(t+h-s) - E_i(t-s)}{h} u(s-r_{ij})ds + \\ &\quad + \int_{t}^{t+h} \frac{E_i(t+h-s)}{h} u(s-r_{ij})ds\end{aligned}$$

Voor $t > 0$ is de functie $E_i(t)$ continu differentieerbaar. We laten $h \downarrow 0$ en dan geldt:

$$\begin{aligned}\frac{Ku(t+h) - Ku(t)}{h} &= E_i(0)u(t-r_{ij}) - \frac{1}{\tau_i}\int_{t_0}^{t} \frac{dE_i}{dt}(t-s)u(s-r_{ij})ds \\ &= u(t-r_{ij}) - \frac{1}{\tau_i}\int_{t_0}^{t} E_i(t-s)u(s-r_{ij})ds \\ &= u(t-r_{ij}) - \frac{1}{\tau_i}Ku(t).\end{aligned}$$

4.2.2 Geval $h < 0$

Laat nu $h < 0$. Dan geldt dat $|h| = -h$. Er volgt:

$$\begin{aligned}\frac{Ku(t+h) - Ku(t)}{h} &= \int_{t_0}^{\infty} \frac{E_i(t-|h|-s - E_i(t-s)}{h} u(s-r_{ij})ds \\ &= \int_{t_0}^{t} \frac{E_i(t-|h|-s - E_i(t-s)}{h} u(s-r_{ij})ds \\ &= \int_{t-|h|}^{t} -\frac{E_i(t-s)}{h} u(s-r_{ij})ds + \\ &\quad + \int_{t_0}^{t-|h|} \frac{E_i(t-|h|-s) - E_i t-s}{h} u(s-r_{ij})ds\end{aligned}$$

Introduceren van $\sigma = t - s, \tilde{t} = t - |h|, \tilde{h} = |h|$, dan levert dit ons:

$$\begin{aligned}
\frac{Ku(t+h) - Ku(t)}{h} &= \int_{|h|}^{0} \frac{E_i(\sigma)}{h} u(t - \sigma - r_{ij}) d\sigma + \\
&\quad + \int_{t_0}^{\tilde{t}} \frac{E_i(t-s) - E_i(\tilde{t} + |h| - s)}{h} u(s - r_{ij}) ds \\
&= -\int_{0}^{|h|} \frac{E(\sigma)}{h} u(t - \sigma - r_{ij}) d\sigma + \\
&\quad + \int_{t_0}^{\tilde{t}} \frac{E_i(\tilde{t} + |h| - s) - E_i(t-s)}{|h|} u(s - r_{ij}) ds \\
&= \frac{1}{|h|} \int_{0}^{|h|} E(\sigma) u(t - \sigma - r_{ij}) d\sigma + \\
&\quad + \int_{t_0}^{\tilde{t}} \frac{E_i(\tilde{t} + \tilde{h} - s) - E_i(t-s)}{\tilde{h}} u(s - r_{ij}) ds
\end{aligned}$$

Laten nu we $h \uparrow 0$, ofwel $\tilde{h} \downarrow 0$, dan volgt:

$$\begin{aligned}
\frac{Ku(t+h) - Ku(t)}{h} &= E(0) u(t - r_{ij}) + \int_{t_0}^{t} \frac{dE_i}{dt}(t-s) u(s - r_{ij}) ds \\
&= u(t - r_{ij}) - \frac{1}{\tau_i} Ku(t)
\end{aligned}$$

Er volgt dus dat:

$$\lim_{h \to 0} \frac{Ku(t+h) - Ku(t)}{h} = \frac{d}{dt} Ku(t) = u(t - r_{ij}) - \frac{1}{\tau_i} Ku(t)$$

Of ook:

$$(\frac{d}{dt} + \frac{1}{\tau_i}) Ku(t) = u(t - r_{ij}).$$

4.3 Het wiskundige model

Gebruiken we het laatste resultaat bij (3) dan zien we het volgende:

$$\begin{aligned}(\frac{d}{dt}+\frac{1}{\tau_i})V_i &= (\frac{d}{dt}+\frac{1}{\tau_i})\sum_j w_{ij}[\int_{t_0}^{t} E_i(t-s)u_j(s)ds] \\ &= \sum_j w_{ij}(\frac{d}{dt}+\frac{1}{\tau_i})[\int_{t_0}^{t} E_i(t-s)u_j(s)ds] \\ &= \sum_j w_{ij}(\frac{d}{dt}+\frac{1}{\tau_i})Ku_j(t) \\ &= \sum_j w_{ij}u_j(t-r_{ij})\end{aligned}$$

Het resultaat is:

$$\dot{V_i} = \sum w_{ij}S_j(V_j(t-r_{ij})) - \frac{1}{\tau_i}V_i \tag{5}$$

Opmerking: in [4] staat een verkeerde vergelijking

$$\tau_i\dot{V_i} + V_i = \sum_j w_{ij}u_j(t-r_{ij}),$$

ofwel

$$\dot{V_i} = \frac{1}{\tau_i}\sum_j w_{ij}u_j(t-r_{ij}) - \frac{1}{\tau_i}V_i.$$

Hierbij staat een extra term τ_i voor het somteken. Een manier om dit in te zien gaat als volgt: Stel de vergelijking is wel correct, dan moet er gelden:

$$(\frac{d}{dt}+\frac{1}{\tau_i})Ku(t) = \frac{1}{\tau_i}u(t-r_{ij})$$

Oftewel

$$(\tau_i\frac{d}{dt}+Id)Ku(t) = u(t-r_{ij})$$

Neem nu $u \equiv 1$. Dan volgt:

$$\begin{aligned}Ku(t) &= \int_{t_0}^{t} e^{(-\frac{(t-s)}{\tau_i}}ds \\ &= [\tau_i e^{-\frac{(t-s)}{\tau_i}}]_{t_0}^{t} \\ &= \tau_i - \tau_i e^{-\frac{t-t_0}{\tau_i}}, \\ \frac{d}{dt}Ku(t) &= e^{-\frac{t-t_0}{\tau_i}}.\end{aligned}$$

Vullen we dit in, dan krijgen we:

$$\begin{aligned}(\tau_i \frac{d}{dt} + Id)Ku(t) &= (\tau_i \frac{d}{dt} + Id)(\tau_i - \tau_i e^{-\frac{t-t_0}{\tau_i}}) \\ &= \tau_i e^{-\frac{t-t_0}{\tau_i}} + \tau_i - \tau_i e^{-\frac{t-t_0}{\tau_i}} \\ &= \tau_i \neq 1\end{aligned}$$

Terwijl, als we het invullen in onze vergelijking, dan krijgen we:

$$\begin{aligned}(\frac{d}{dt} + \frac{1}{\tau_i})Ku(t) &= (\frac{d}{dt} + \frac{1}{\tau_i})(\tau_i - \tau_i e^{-\frac{t-t_0}{\tau_i}}) \\ &= e^{-\frac{t-t_0}{\tau_i}} + 1 - e^{-\frac{t-t_0}{\tau_i}} \\ &= 1\end{aligned}$$

5 Analyse van een netwerk van neuronen

In dit hoofdstuk wordt analyse gedaan van een netwerk van neuronen. Dat wil zeggen, we kijken of er evenwichten bestaan, we doen bifurcatie-analyse en we gaan op zoek naar verschillen tussen de differentiaalvergelijking en de delayvergelijking. We gaan hierbij uit van model (5), waarbij we veronderstellen dat de karakteristieke tijd τ_i voor iedere neuron gelijk is: $\tau_i = \tau$

5.1 Een dimensievrije vergelijking

De V_i in vergelijking (5) heeft de eenheid Volt, en is dus niet dimensievrij. We maken deze dimensievrij door een afbeelding

$$V_i \mapsto \tilde{V}_i = \frac{V_i}{v_i}, \qquad t \mapsto \tilde{t} = \frac{t}{\tau}$$

waarbij v_i ook dimensie Volt heeft, maar niet van de tijd afhangt Deze v_i kunnen we zelf kiezen. Zoals we zo zullen zien, heeft dit een extra voordeel. Onze nieuwe, "dimensievrije"vergelijking wordt:

$$\frac{v_i}{\tau}\frac{d\tilde{V}_i}{d\tilde{t}} + \frac{v_i}{\tau}\tilde{V}_i = \sum_j w_{ij} S_j(v_j \tilde{V}_j(\tilde{t} - \frac{r_{ij}}{\tau})).$$

Links en rechts delen door v_i en introduceren van

$$\begin{aligned} \tilde{S}_j(x) &= S_j(v_j x) \\ \tilde{r}_{ij} &= \frac{r_{ij}}{\tau} \\ \tilde{w}_{ij} &= w_{ij}\tau \end{aligned}$$

geeft ons nu

$$\frac{d\tilde{V}_i}{d\tilde{t}} + \tilde{V}_i = \sum_j (\frac{\tilde{w}_{ij}}{v_i})\tilde{S}_j(\tilde{V}_j(t - \tilde{r}_{ij})). \tag{6}$$

Voor het gemak laten we alle tildes weg.
In het geval van een netwerk van n neuronen, kunnen we de elementen $A = w_{ij}$ beschouwen als een $n \times n$-matrix. In de nieuwe dimensievrije vergelijking hebben we een nieuwe matrix $W = \frac{w_{ij}}{v_i}$. Mits de diagonaalelementen van de matrix A alle niet gelijk zijn aan 0, dan kunnen we kiezen $v_i = \pm w_{ii}$ en zo krijgen we dat de diagonaalelementen van matrix W alle bestaan uit ± 1. Als alternatief kunnen we ook nemen $v_i = \max\{|w_{ij}|, j = 1, 2, \ldots, n\}$. We kiezen er voor om te nemen $v_i = \pm w_{ii}$, it maakt het doen van analyse van bijvoorbeeld het geval dat $n = 2$ gemakkelijker, doordat het aantal parameters met n gereduceerd wordt. We veronderstellen verder dat voor iedere neuron de functie S hetzelfde is: $S_j = S$.

5.2 Bestaan van evenwichten

We zoeken in dit hoofdstuk het bestaan van evenwichten van de vergelijking (5) voor zowel $r_{ij} = 0$ voor alle i, j, als voor $r_{ij} \neq 0$. Deze twee behandelen we apart.

5.2.1 Zonder delay

We bekijken het geval dat $r_{ij} = 0$ voor alle i, j. In dit geval is vergelijking (6) een differentiaalvergelijking. We nemen aan dat $n = 2$. Voor een evenwichtswaarde (V_1^*, V_2^*) geldt dat $\dot{V}_i(t) = 0$. Gebruiken we dit in vergelijking (6) dan krijgen we

$$V_1^* = w_{11}S(V_1^*) + w_{12}S(V_2^*)$$
$$V_2^* = w_{21}S(V_1^*) + w_{22}S(V_2^*)$$

Schrijven we

$$\underline{S}(\underline{V}) = \begin{pmatrix} S(V_1) \\ S(V_2) \end{pmatrix},$$
$$W = \begin{pmatrix} \pm 1 & \frac{w_{12}}{v_1} \\ \frac{w_{21}}{v_2} & \pm 1 \end{pmatrix} = \begin{pmatrix} \pm 1 & d_1 \\ d_2 & \pm 1 \end{pmatrix}$$

dan krijgen we

$$V^* = W\underline{S}(\underline{V}^*).$$

We moeten dus op zoek naar een fixed punt van $W\underline{S} = \Psi$.

Voor het vinden van een fixed punt merken we op dat voor de functie S_i het beeld geheel in het interval $[0, S_{max}]$ ligt, voor alle i. Er geldt dus dat $\underline{S}(\underline{x}) \in \{\underline{x} | 0 < x_i < S_{max}\}$.
Zij B het blok zodanig dat

$$B = \{\lambda e_1 + \mu e_2 | 0 \leq \lambda, \mu \leq S_{max}\}$$

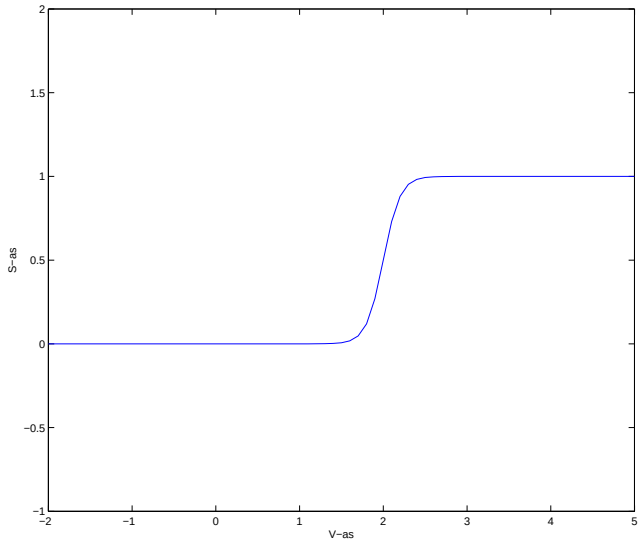
Uit het feit dat

$$We_1 = \begin{pmatrix} \pm 1 \\ d_2 \end{pmatrix}, \qquad We_2 = \begin{pmatrix} d_1 \\ \pm 1 \end{pmatrix},$$

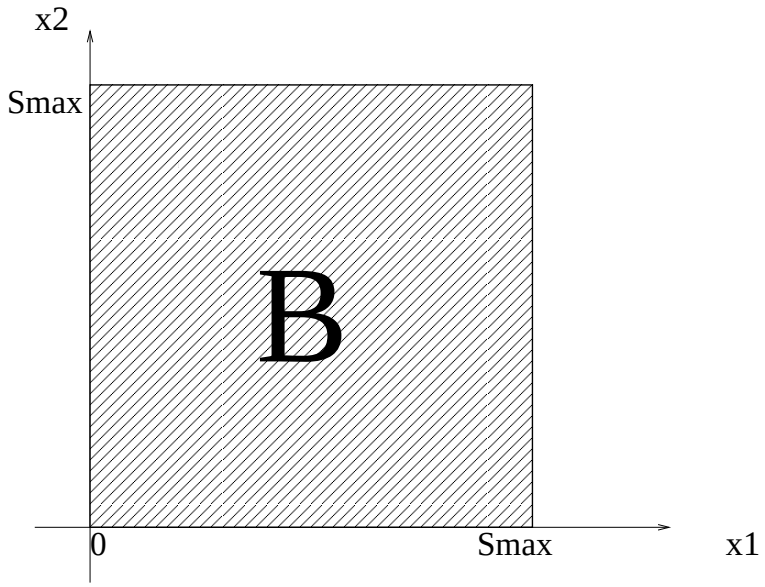
volgt dat

$$W(B) = \{\lambda \begin{pmatrix} \pm 1 \\ d_2 \end{pmatrix} + \mu \begin{pmatrix} d_1 \\ \pm 1 \end{pmatrix} | 0 \leq \lambda, \mu \leq S_{max}\}$$

De functie S beeldt $\mathbf{R}^2$, dus in het bijzonder $W(B)$, af in B. Dus $\Psi : W(B) \rightarrow W(B)$. Ψ is continu, en $W(B)$ is compact (want deze is begrensd en gesloten).



Figuur 12: Functionele vorm van S met $S_{max} = 1, V_T = 2, V_S = 0.1$

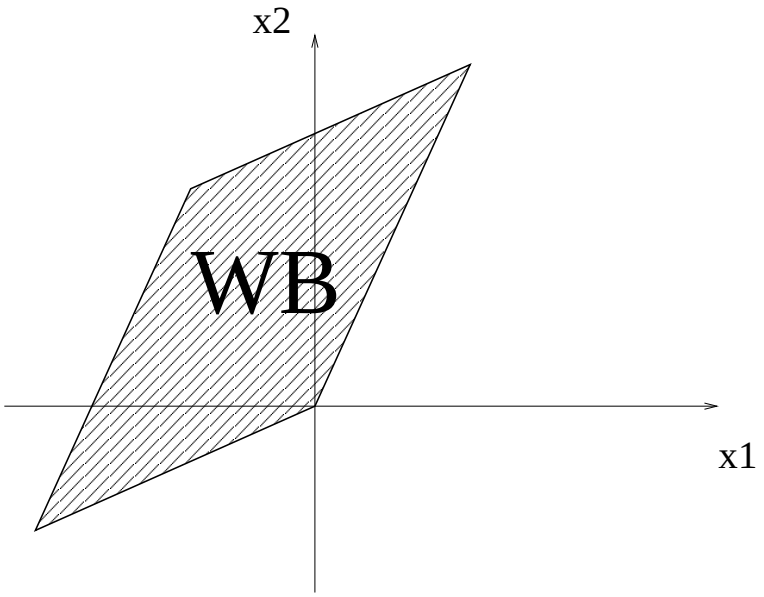


Figuur 13: Deelverzameling $B \subset \mathbf{R}^2$

(Zie Figuur 14). Volgens de fixed-punt stelling van Brouwer (zie [5], theorem 4.2.5, p.119), geldt nu dat er een fixed punt is. We merken hierbij op dat, indien det $W \neq 0$, dan is dit niet een triviaal fixed punt, i.e. $V^* = (0, 0)$, omdat $S(V) > 0$ voor alle V.

Stelling 1 *Zij C een compacte convexe deelverzameling van een eindig dimensionale ruimte en $f : C \to C$ een continue afbeelding. Dan bestaat er een punt $\tilde{x} \in C$ zodat $f(\tilde{x}) = \tilde{x}$*

Voor een bewijs, zie [5].



Figuur 14: Deelverzameling $W(B) \subset \mathbf{R}^2$

5.2.2 Voor de delayvergelijking

Voor de delayvergelijking hebben we

$$\dot{V}_i(t) + \frac{1}{\tau} V_i(t) = \sum_j w_{ij} S(V_j(t - r_{ij}))$$

waarbij niet alle r_{ij} gelijk aan 0. Voor het gemak schrijven we deze om naar

$$\begin{aligned} \dot{V}_i(t) &= -\frac{1}{\tau} V_i(t) + \sum_j w_{ij} S(V_j(t - r_{ij})) \\ &= F(V_i(t), V_i(t - r_{i1}), \ldots, V_i(t - r_{in})) \end{aligned}$$

Stel dat er een evenwicht V_i^* bestaat, dan moet daarvoor gelden dat $\dot{V}_i^*(t) = 0$. Hieruit volgt dat $V_i(t) = V_i(t - r_{ij})$ voor alle i, j. Er geldt dus dat

$$\begin{aligned}\dot{V}_i(t) &= F(V_i(t), V_i(t - r_{i1}), \ldots, V_i(t - r_{in})) \\ &= F(V_i(t), V_i(t), \ldots, V_i(t)) = 0\end{aligned}$$

Merk nu op dat de rechter-term in feite weer de differentiaalvergelijking is, dat wil zeggen de vergelijking zonder delay. Conclusie: V^* heeft dezelfde waarde, zonder of met delay.

5.3 Het oplossen van de delayvergelijking door middel van successieve integratie

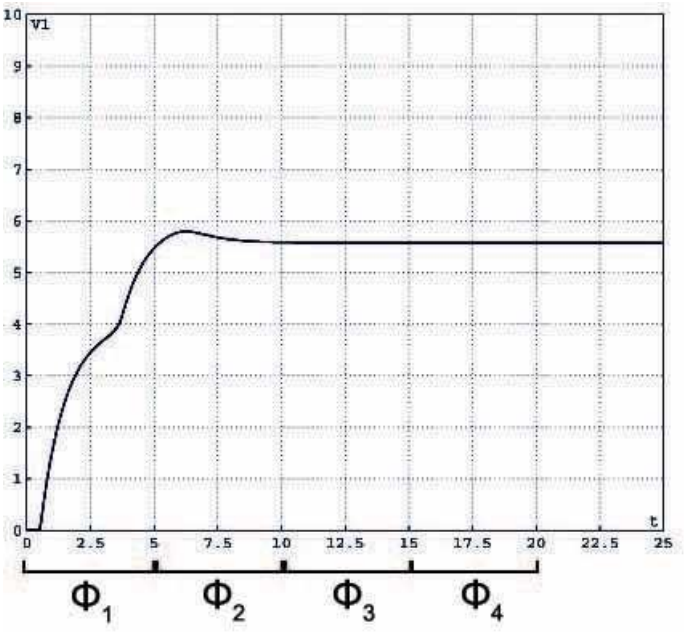
In deze paragraaf wordt een manier uitgelegd om de delayvergelijking op te lossen. Deze manier, die successieve integratie wordt genoemd, gebruiken we ook om met matlab numeriek de oplossing te bepalen.

We gaan er vanuit dat $r_{ij} \neq 0$ voor ten minste één paar i, j: $\delta = \max_{i,j}\{r_{ij}\} > 0$. De oplossing op het interval $[-\delta, 0]$ is gegeven als beginvoorwaarde voor de delayvergelijking. Dit is een functie $\phi_0 \in C([-\delta, 0], \mathbf{R}^n)$. Deze ϕ_0 kunnen we invullen in vergelijking (5):

$$\dot{V}_i = \sum w_{ij} S_j(\phi_0(x)) - \frac{1}{\tau_i} V_i$$

en hiermee vinden we de oplossing op het interval $[0, \delta]$. Deze oplossing noemen we ϕ_1, en die kunnen we weer invullen in vergelijking (5). Uiteindelijk geeft dit ons de totale oplossing ϕ. In de volgende paragraaf zullen we zien dat dit een dynamisch systeem in $X = C([-\delta, 0], \mathbf{R}^n)$ genereert.

Het feit dat voor $0 < t < 1$ de oplossing zo goed als horizontaal loopt, wordt verklaard in paragraaf 6.1.



Figuur 15: Oplossingen $\phi_1, \phi_2, \phi_3, \phi_4$ voor $\delta = 5$

5.4 Een standpunt van oneindig-dimensionale dynamische systemen

We zetten hier het idee van een dynamisch systeem uit, eindig en oneindig dimensionaal.

Voordat we beginnen met een formele definitie, is het handig om eerst uit te leggen wat een dynamisch systeem inhoudt. Voor een dynamisch systeem hebben we een toestandsruimte X en de tijdruimte T, vaak: $T = \mathbf{R}$ of $T = [0, \infty)$. We beschouwen de tijd dus als iets dat continu is. In het algemeen kan X een metrische ruimte zijn. In ons geval is X altijd een genormeerde vectorruimte. Hierdoor is X ook een metrische ruimte met $d(x, y) = ||x - y||$. Voor iedere tijdstip $t \in T$ bestaat er een toestand $x(t) \in X$.

We nemen aan dat er een beginwaarde $x_0 = x(t_0)$ op tijd t_0 is die samen met een functie $f : T \to X, t \mapsto x(t)$, de toestanden voor alle $t \in T$ uniek bepaalt. Dat wil zeggen dat er een functie $\phi : T \times T \times X \to X$ is zodat

$$x(t) = \phi(t; t_0, x_0).$$

In principe willen we ook dat de keuze van t_0 niet uitmaakt. Dit maakt het geheel een stuk eenvoudiger. Hierdoor kunnen we schrijven

$$\phi(t, t_0, x_0) = \phi(t - t_0, x_0)$$

Om het nog eenvoudiger te maken kiezen we $t_0 = 0$ en we hebben

$$\phi(t - t_0, x_0) = \phi(t; x_0) = \phi^t(x_0)$$

Anders gezien: voor iedere $t \in T$ hebben we een afbeelding ϕ^t. Zo introduceren we Φ als de verzameling afbeeldingen: $\Phi = \{\phi^t\}_{t \in T}$

In principe hoeft T niet altijd gelijk te zijn aan $\mathbf{R}$. In het geval dat we T als discrete tijd beschouwen, kunnen we $T = \mathbf{Z}$ nemen. In sommige gevallen kunnen we alleen de toekomst beschouwen en niet het verleden, in zulke gevallen hebben we $T = \mathbf{R}_+, \mathbf{Z}_+$.

Voor Φ hebben we twee belangrijke eigenschappen:

$$\phi^0(x) = x \tag{7}$$

$$\phi^{t+s}(x) = \phi^t(\phi^s(x)) \tag{8}$$

Eigenschap (7) betekent dat x de gegeven toestand is op tijdstip $t = 0$. Eigenschap (9) betekent dat de toestand na tijd $t + s$ (beginnende op een toestand x) hetzelfde is als de toestand na tijd t, beginnende met toestand $\phi^s(x)$. Uit de laatste eigenschap volgt dat Φ een semigroep is.

Definitie 1 *Een dynamisch systeem in een genormeerde vectorruimte X is een familie van afbeeldingen $\Phi(t)_{t \in T} : X \to X$ continu, zodat voldaan wordt aan* (7) *en* (9) *voor alle $t, s \in T$ en zó dat de afbeelding $t \mapsto \Phi(t)x : T \to X$ continu is voor alle $x \in X$*

5.4.1 Voorbeelden

Voorbeeld 1 *Iteratie van afbeeldingen (Mapiteratie)*

Eerst een simpel voorbeeld van een dynamisch systeem. Neem $X = \mathbf{R}$ met de gebruikelijke norm, en $T = \mathbf{Z}_+$. Zij $f : X \to X$ een continue functie. Zet nu $\phi^0 = Id$ en

$$\phi^k = f \circ \phi^{k-1} = \underbrace{f \circ f \circ \ldots \circ f}_{k \text{ keer}}$$

Dan is de familie van afbeeldingen $\Phi(k)_{k \in T}$ samen met X een dynamisch systeem.

Voorbeeld 2 *Beginwaardeprobleem*

Beschouw een van beginwaardeprobleem

$$\begin{aligned} \dot{x} &= f(x), \qquad x \in E \subset \mathbf{R}^n \\ x(0) &= x_0 \end{aligned} \tag{9}$$

met $f : \mathbf{R}^n \to \mathbf{R}^n$

Stelling 2 *Zij $E \subset \mathbf{R}^n$ zodat $x_0 \in E$ en neem aan dat $f \in C^1(E)$. Dan is er een $a > 0$ zodat het beginwaardeprobleem (9) een unieke oplossing $x(t)$ heeft op het interval $[-a, a]$*

Voor een bewijs, zie [3]. Laat nu $x(t) = \phi(t, x_0) = \phi^t(x_0)$, dan is $\Phi(t)$ samen met E een dynamisch systeem, wanneer voor alle $x_0 \in E$ de functie ϕ goed gedefinieerd is.

Voorbeeld 3 *De delayvergelijking*

In de vorige paragraaf zagen we dat we de delayvergelijking kunnen oplossen door middel van successieve integratie. Hiervoor beginnen we met een beginwaarde $\phi^{(0)} \in C([-\delta, 0], \mathbf{R}^n)$. Deze invullen levert een $\phi^{(1)} \in C([0, \delta], \mathbf{R}^n)$. Deze kunnen we transleren naar het interval $[-\delta, 0]$. In feite hebben we voor iedere $t \in \mathbf{R}_+$ een deeloplossing $\phi^{(t)}$: de beperking van de hele oplossing tot het interval $[-\delta + t\delta, t\delta]$. Deze kunnen we transleren naar het interval $[-\delta, 0]$. Dit geeft ons een dynamisch systeem in toestandsruimte $X = C([-\delta, 0], \mathbf{R}^n)$, beschreven door

$$\Phi(t)\phi_0 = \phi_t.$$

Voor een $\theta \in [-\delta, 0]$ geldt:

$$\phi_t(\theta) = \phi(t + \theta)$$

waarbij ϕ de gehele oplossing is: $\phi : [-\delta, \infty) \to \mathbf{R}^n$

Een stationair punt van $(\Phi(t))_{t\geq 0}$ is een punt $\psi \in C([-\delta, 0], \mathbf{R}^n)$ zodanig dat:

$$\Phi(t)\psi = \psi$$

voor alle t. Het is à priori niet duidelijk wat voor een element dit is uit $C([-\delta, 0], \mathbf{R}^n)$. Het blijkt dat ψ een constante functie is. Immers, voor $\theta \in [-\delta, 0]$ geldt:

$$\begin{aligned}\psi(\theta) &= \psi(-\delta + (\theta + \delta)) \\ &= \psi_{\theta+\delta}(-\delta) = [\Phi(\theta + \delta)\psi](-\delta) \\ &= \psi(-\delta)\end{aligned}$$

6 Numerieke analyse

In dit hoofdstuk wordt numerieke analyse van het wiskundig model voor een netwerk van neuronen gedaan. Dat bestaat uit twee gedeelten: Bifurcatieanalyse van het model zonder delay, wat gedaan wordt met CONTENT, en het oplossen van het model met delay onder andere door middel van successieve integratie. Dit laatste is gedaan met Matlab.

Bij deze numerieke analyse stimuleren we neuron 2, en alleen deze, met een prikkel. Deze prikkel draagt de letter I van *Input*. De vergelijkingen die we analyseren zijn dus:

$$\dot{V_1} = \sum_j w_{1j} S_j(V_j(t - r_{1j})) - \frac{1}{\tau_1} V_1$$
$$\dot{V_2} = \sum_j w_{2j} S_j(V_j(t - r_{2j})) - \frac{1}{\tau_2} V_2 + I$$

6.1 Bifurcatieanalyse van het model, zonder delay

Tijdens bifurcatieanalyse van het model zonder delay hebben we ons beperkt tot het geval dat $n = 2$. In dit geval is de bifurcatieanalyse ook twee-dimensionaal, en kunnen we er dus grafieken bij maken. We bekijken de twee gevallen:

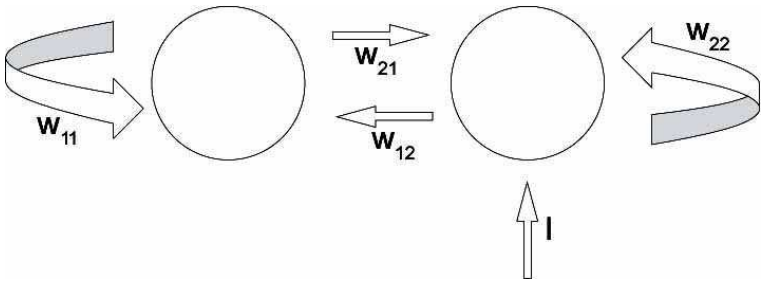
$$\begin{pmatrix} 2 & w_{12} \\ w_{21} & 2 \end{pmatrix}, \qquad \begin{pmatrix} 2 & w_{12} \\ w_{21} & -2 \end{pmatrix},$$

en we doen bifurcatieanalyse over de parameters w_{12}, w_{21}. Verder zetten we parameters:

$$S_{max} = 1$$
$$V_T = 4$$
$$\tau = 1$$
$$I = 10$$

In het geval dat $w_{11} = w_{22} = 2$ hebben we gebruikt $V_S = 0.3$, en in het geval $w_{11} = -w_{22} = 2$ hebben we gebruikt $V_S = 0.1$. De reden waarom we $w_{11} = w_{22} = \pm 2$ in plaats van ± 1 is vanwege numerieke problemen (overflows) die we tijdens het onderzoek zijn tegengekomen. Dit is ook de reden waarom $V_S = 0.3$ in het eerste geval.

We beginnen met het geval $w_{11} = -w_{22} = 2$. Voor de eenvoud nemen we $w_{12} = 0, w_{21} = 0$. Samengevat:

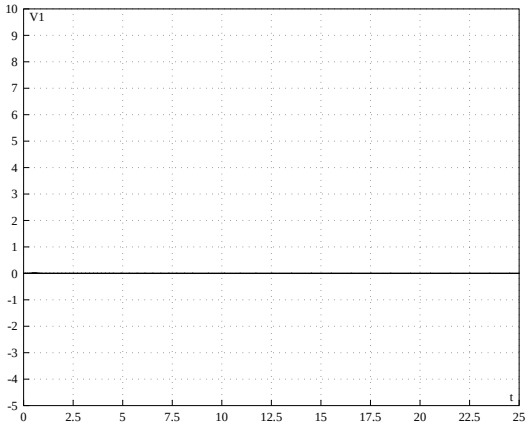


Figuur 16: Schematische weergave van het netwerk van 2 neuronen

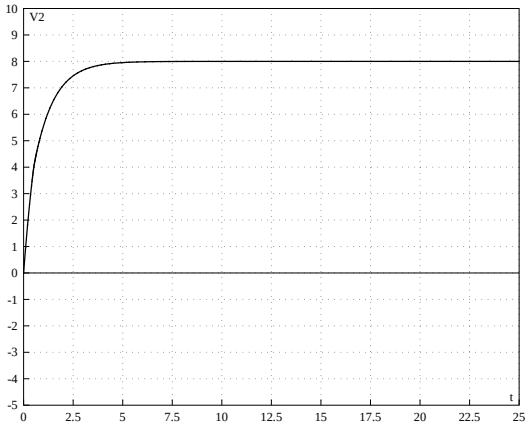
parameter	*waarde*
w_{12}	0
w_{21}	0
S_{max}	1
V_T	4
V_S	0.1
τ	1
I	10

In dit geval zou moeten gelden dat $V_1(t) = 0$ voor alle t. Immers: omdat $w_{21} = 0$ weten we dat er geen pulsen in de richting van neuron 1 worden geschoten. De potentiaal van neuron 2 zou richting de waarde 8 moeten gaan. Immers: deze krijgt een waarde van 10 mee van buitenaf, en inhibeert met waarde 2. Nettowaarde is dus $10 - 2 = 8$ (zie Figuur 17, 18).

Bij de bifurcatiediagrammen die volgen gebruiken we notatie $w2$ en $w3$. Er geldt hier: $w2 = w_{12}, w3 = w_{21}$.

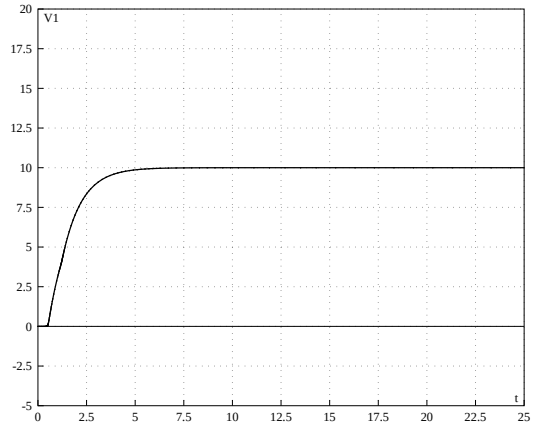


Figuur 17: Verloop van potentiaal V_1 tegen de tijd

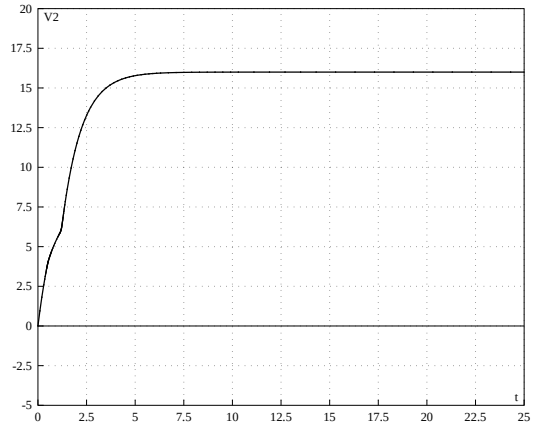


Figuur 18: Verloop van potentiaal V_2 tegen de tijd

Om neuron 1 wel in verbinding te laten staan met neuron 2 kiezen we de volgende parameterwaarden: $w_{12} = w_{21} = 8$. Met CONTENT plotten we vervolgens de oplossingen: Het ziet er naar uit dat voor $0 < t < 1$ de oplossing



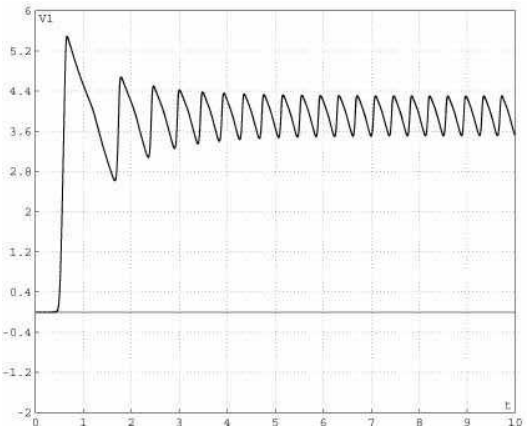
Figuur 19: Verloop van potentiaal V_1 tegen de tijd



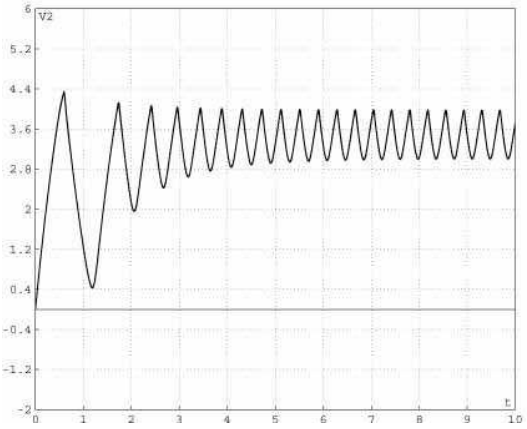
Figuur 20: Verloop van potentiaal V_2 tegen de tijd

vrijwel horizontaal loopt, en daarna in een enkel punt niet differentieerbaar is (zie Figuur 19). Dit laatste is echter niet het geval. Waarom de oplossing pas rond $t = 1$ (en niet $t = 0$) richting het stabiele evenwicht gaat is als volgt te verklaren: de potentiaal V_1 van neuron 1 stijgt pas zodra de potentiaal V_2 van neuron 2 boven de grenswaarde $V_T = 4$ uitkomt. Dit is vanwege de functionele vorm van S_j (Figuur 11). Hetzelfde is waar te nemen bij Figuur 21 en Figuur 15.

Om te laten zien dat er niet alleen stabiele maar ook onstabiele evenwichten zijn, kiezen we $w_{12} = 50, w_{21} = -15$. Met CONTENT plotten we de oplossingen:

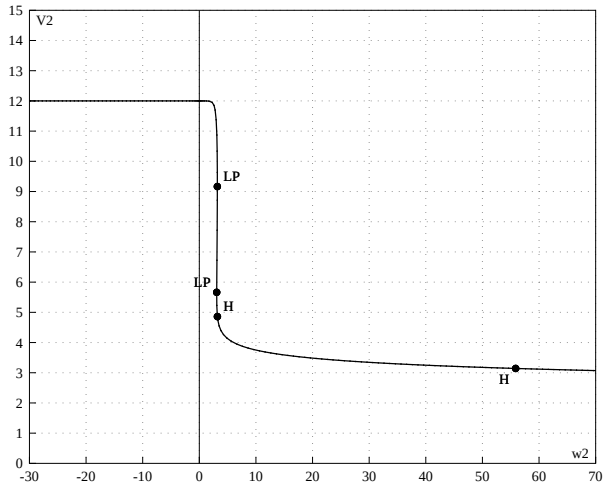


Figuur 21: Verloop van potentiaal V_1 tegen de tijd



Figuur 22: Verloop van potentiaal V_2 tegen de tijd

Met CONTENT kunnen we, indien we w_{21} vast houden, numeriek de waarde van w_{12} bepalen waar het omslagpunt van stabiel evenwicht naar onstabiel evenwicht zit.



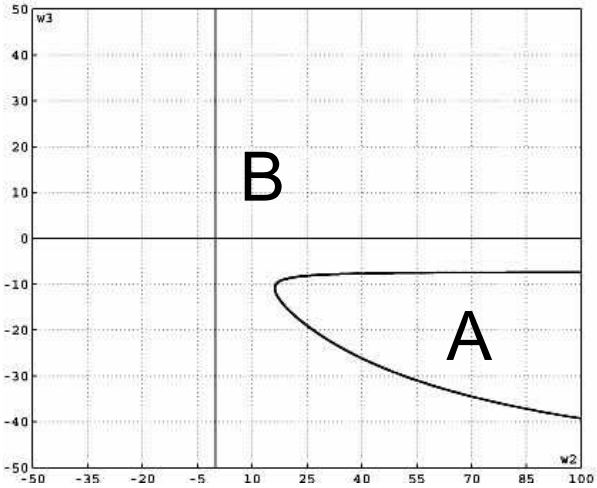
Figuur 23: Voor iedere waarde w_{12} is er een vast punt, te zien aan de grafiek

Rechts in de figuur (Figuur 23) is op de grafiek de letter H ($w_{12} \approx 56$) te zien. Deze staat voor Hopf-bifurcatie. Uit numerieke analyse zien we dat links van deze letter H het evenwichtspunt stabiel is, en rechts ervan is deze onstabiel.

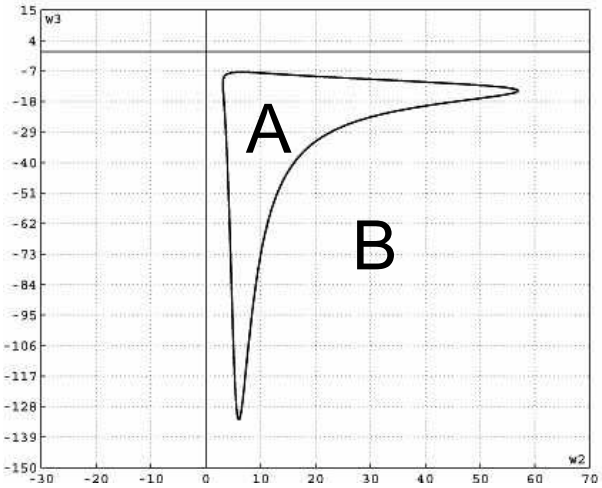
Met CONTENT kunnen we nu zowel w_{12} als w_{21} laten variëren, en zo in het w_{12}, w_{21}-diagram een grafiek met Hopf-bifurcaties maken. We zien dat deze grafiek de $\mathbf{R}^2$ in twee gebieden A en B opdeelt. Zie de figuur (Figuur 24). In gebied A is er een stabiel evenwicht, in gebied B een onstabiel evenwicht.

In het geval dat $w_{11} = w_{22} = 2$ hebben we dezelfde bifurcatieanalyse gedaan, en zijn we gekomen op de volgende figuur (Figuur 25). In gebied A is er een stabiel evenwicht, in gebied B een onstabiel evenwicht.

Nogmaals, er geldt hier dat: $w2 = w_{12}, w3 = w_{21}$.



Figuur 24: Bifurcatie-diagram, $w_{11} = -w_{22} = 2$



Figuur 25: Bifurcatie-diagram, $w_{11} = w_{22} = 2$

6.2 Numeriek oplossen van de delayvergelijking

Met matlab hebben we drie verschillende M-files gemaakt, die alle drie een delayvergelijking kunnen oplossen. Toelichting bij de M-files is te vinden in de volgende paragraaf.

Tijdens dit onderzoek zijn we er vanuit gegaan dat de delay van neuron i naar neuron j, $i \neq j$ altijd een vaste waarde δ is. Verder hebben we dat de delay van neuron i naar neuron i gelijk is aan 0. De M-files zijn zo geschreven dat we ook kunnen kiezen voor $\delta = 0$, zodat we het stelsel van differentiaalvergelijkingen terug krijgen. De eerste controle is dan natuurlijk dat de oplossingen in Matlab overeenkomen met de oplossingen in CONTENT. Dit is inderdaad het geval.

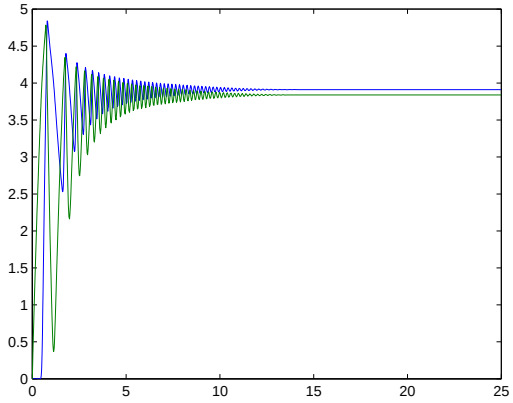
Het voordeel van matlab ten opzicht van CONTENT is, dat we zeer gemakkelijk het aantal neuronen n kunnen veranderen. Matlab verandert de het aantal vergelijkingen dan automatisch mee. Bij CONTENT moesten we heel de vergelijking van het stelstel van differentiaalvergelijkingen opnieuw uittypen, wat een enorm extra werk is.

Het doel van de scriptie is te weten te komen of er verschillen zijn tussen het model met delay en het model zonder delay. Met matlab kunnen we nu kijken of er een verschil is in stabiliteit van de evenwichten op het moment dat we de delay variëren. Het lijkt praktisch om dan de parameterwaarden w_{12} en w_{21} dan dichtbij de grafiek van het bifurcatiediagram te kiezen.

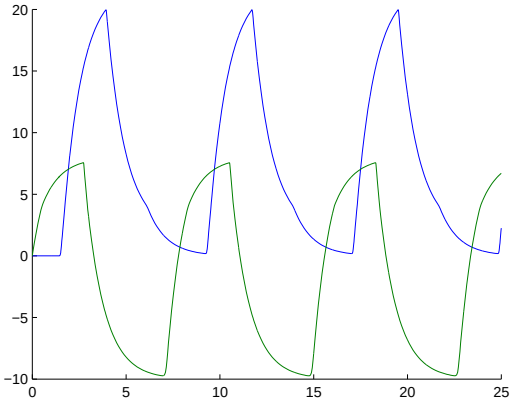
Het resultaat is dat er inderdaad verschil is aan te tonen. Voor het geval dat $w_{11} = -w_{22} = 2$ hebben we de volgende parameters gekozen:

parameter	*waarde*
w_{12}	20
w_{21}	20
S_{max}	1
V_T	4
V_S	0.1
τ	1
I	10

Voor het model zonder delay krijgen we een stabiel evenwicht:V_1 en V_2 worden uiteindelijk constant. (Figuur 26). Het model met delay $\delta = 1$ en zelfde parameterwaarden geeft Figuur 27. Hier zijn twee onstabiele evenwichten waar te nemen. Hier oscilleren V_1 en V_2. Voor zowel met delay als zonder delay hebben we beginwaarden $V_1(0) = 0, V_2(0) = 0$.



Figuur 26: Oplossing van het model zonder delay



Figuur 27: Oplossing van het model met delay

6.3 Toelichting bij de code

In de Bijlage hieronder staan drie verschillende m-files, alle drie in staat het model met delay op te lossen. De eerste twee maken gebruik van successieve integratie, terwijl de laatste gebruik maakt van een ingebouwde matlab-dde-tool (dde staat voor delay differential equations). Hieronder wordt uitgelegd wat het verschil is tussen de eerste twee m-files.

De 'history'-functie $\phi_t \in C([-\delta + t, t], \mathbf{R}^n)$ wordt in matlab opgeslagen als een vector *ydel* van lengte 51 en breedte n. De lengte 51 van de matrix heeft de volgende achterliggende gedachte: nemen we delay $\delta = 1$, dan kunnen we het interval opdelen in precies 50 partities van gelijke lengte.
Het eerste probleem waar we tegen aan liepen was het volgende: Op het moment dat we op tijdstip $t = \theta$ wilden voorwaarts integreren, moesten we binnen de functie *dydt* de waarde op tijdstip $\theta - \delta$ uit de vector *ydel* halen. Dit ging nogal lastig, dus hebben we dat op de volgende manier omzeilt: Zodra $\frac{1}{50}^e$ deel van de nieuwe oplossing bekend is, wordt er direct een nieuwe vector *ydel* gemaakt. In dit geval hoeven we nergens de waarde op tijdstip $\theta - \delta$ te weten, maar alleen de waarde $-\delta$. Dit is altijd de bovenste rij van de vector *ydel* en die kunnen we expliciet aanroepen binnen de functie *dydt*. Het komt er dus op neer dat, voordat we de eerste nieuwe functie $\phi_1 \in C([0, \delta], \mathbf{R}^n)$ hebben verkregen, matlab al 50 keer een ODE-tool heeft aangeroepen. We noemen de eerste m-file dus ook de trage manier van oplossen

Later in het onderzoek wisten we hoe we 'normaal' successieve integratie konden toepassen in matlab, en zo hebben we uit de eerste m-file een snellere gemaakt. Het probleem waar we nu tegenaan liepen was dat matlab niet in staat was de functiewaarden op de rand van een interval te bepalen. Dit hebben we omzeilt door de de waarde van het randpunt te vergroten of verkleinen met $100 * \text{eps}$.

Aangezien de drie m-files alle in staat zijn het model met delay op te lossen, willen we natuurlijk dat oplossingen overeenkomen (dat wil zeggen: de oplossing van de eerste m-file moet hetzelfde zijn als de oplossing van de tweede m-file en de derde m-file). Dit is gecontroleerd, en het klopt dat de oplossingen precies hetzelfde zijn.

7 Discussie

We hebben de bouw en werking van een enkele neuron bekeken, en hebben daarmee fasevlak analyse gedaan. We zijn hieruit verder gegaan naar het model van een netwerk van neuronen. Met een netwerk van twee neuronen hebben we fasevlak analyse gedaan, en gekeken of er verschil te zien was tussen het geval dat de delay gelijk was aan 0, en het geval dat de delay niet gelijk was aan 0. We hebben gezien dat er inderdaad verschil te zien is.

We hadden nog kunnen kijken naar het geval dat de input I tijdsafhankelijk is, of hoe de bifurcatiediagrammen veranderen wanneer de waarde van I verandert, of hoe de diagrammen convergeren als $\delta \downarrow 0$.

Verder hebben we nog wel een tool gevonden voor bifurcatie-analyse voor het model met delay. Deze tool heet *DDE-BIFTOOL*. Een artikel over hoe de tool werkt en waar men deze kan verkrijgen staat op
`http://www.cs.kuleuven.ac.be/publicaties/rapporten/tw/TW330.abs.html`.
Het kostte teveel tijd om zelf uit te zoeken hoe de tool werkte.

8 Bijlage

8.1 Bijlage 1: 'trage' manier van oplossen met behulp van successieve integratie

```
%-----------------------------------------------%
%Vanaf hier wordt de ODE-functie aangeroepen. %
%-----------------------------------------------%

function X = oderun(ydel, delay, mat, Input)
%  delay < 1
%  plot komt uiteindelijk vanaf 0 tot 0.02*m*n
n = length(ydel)-1;
m = 5;
l = length(ydel'*ydel);
B = zeros(1, l);
if (delay == 0)
    [t y] = ode23s(@zonderdelay, [0 0.02*m*n], B, [], mat, Input);
    plot(t, y)
else
    P = ydel;
    hold on;
    C = B;
    for k=1:m
        yint = 0 + (k-1):.02:k;
        for j=1:n
```

```
            for z=1:l
                B(z) = P(z*(n+1));
                C(z) = P((z*(n+1)) - n*delay);
            end
            xint = 0+n*(0.02)*(k-1)+0.02*(j-1):.02:1+n*(k-1)*(0.02)+0.02*(j-1);
            sol = ode23s(@metdelay, [0+n*(0.02)*(k-1)+0.02*(j-1),
                1+n*(k-1)*(0.02)+0.02*(j-1)], B, [], C, mat, Input);
            Sxint = deval(sol, xint);
            for i=1:n
                for g=1:l
                ydel(i+(n+1)*(g-1)) = ydel(i+1+(n+1)*(g-1));
                end
            end
            for i=1:l
                ydel((n+1)*i) = Sxint(l+i);
            end
        P = ydel;
        end
        plot(yint, P)
    end
end

%-----------------------------------%
%Hier de ODE-functie ZONDER delay. %
%-----------------------------------%

function dydt = zonderdelay(t,y, mat, Input)
%Aantal neuronen n:
n = length(mat);
%Nieuwe 2-vector:
dydt = zeros(n, 1);
%Parameters:
w = mat; %(inputmatrix)
Smax = 1;
VT = 4;
VS = 0.1;
tau = 1;
% Hier staat de ODE:
for i=1:n
    for j=1:n
      dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(y(j)-VT)/VS))));
    end
    dydt(i) = (dydt(i) - y(i)) / tau + Input(i);
end

%--------------------------------%
```

```
%Hier de ODE-functie MET delay. %
%-------------------------------%

function dydt = metdelay(t,y, ydel, mat, Input)
%   Aantal neuronen n:
n = length(mat);
%   Nieuwe 2-vector aanmaken:
dydt = zeros(n, 1);
%   Parameters:
w = mat;
Smax = 1;
VT = 4;
VS = 0.1;
tau = 1;
%   Hier staat de ODE:
for i=1:n
    for j=1:n
        if (i==j)
            dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(y(j)-VT)/VS))));
        else
            dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(ydel(j)-VT)/VS))));
        end
    end
    dydt(i) = (dydt(i) - y(i)) / tau + Input(i);
end
```

8.2 Bijlage 2: snelle manier van oplossen met behulp van successieve integratie

```
%-----------------------------------------%
%Vanaf hier wordt de ODE-functie aangeroepen. %
%-----------------------------------------%

function X = oderun(delay, mat, input, parm)
m = 25;
l = length(mat);
B = zeros(1, l);
%  controle op juiste invoergegevens
if (l == 0 || length(input) ~= l || length(parm) ~= 4)
    error('verkeerde input!!')
    if (delay ~= 0)
        if (round(m/delay) - (m/delay) ~= 0 && round(m/delay) - (m/delay) ~= 1)
            error('6 / delay moet een geheel getal vormen')
        end
    end
end
if (delay == 0)
    sol = ode23s(@zonderdelay, [0 m], B, [], mat, input, parm);
    yvalue = getfield(sol,'y');
    xvalue = getfield(sol,'x');
    plot(xvalue, yvalue)
else
    sol = B;
    hold on;
    for k=1:(m/delay)
        sol1 = ode23s(@metdelay, [(k-1)*delay, k*delay],
            B, [], delay, mat, input, parm, sol, k);
        sol = sol1;
        yvalue = getfield(sol,'y');
        xvalue = getfield(sol,'x');
        ylength = length(yvalue);
        ywidth = l;
        for i=1:l
            B(i) = yvalue((ylength-1)*ywidth+i);
        end
        plot(xvalue,yvalue)
    end
end

%----------------------------------%
%Hier de ODE-functie ZONDER delay. %
%----------------------------------%
```

```
function dydt = zonderdelay(t,y, mat, input, parm)
%Aantal neuronen n:
n = length(mat);
%Nieuwe 2-vector aanmaken:
dydt = zeros(n, 1);
%Parameters:
w = mat; %(inputmatrix)
Smax = parm(1);
VT = parm(2);
VS = parm(3);
tau = parm(4);
% Hier staat de ODE:
for i=1:n
    for j=1:n
      dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(y(j)-VT)/VS))));
    end
    dydt(i) = (dydt(i) - y(i)) / tau + input(i);
end

%-------------------------------%
%Hier de ODE-functie MET delay. %
%-------------------------------%

function dydt = metdelay(t,y, delay, mat, input, parm, sol, k)
%   De eerste keer geldt C = ydel = (0, 0)
a = isstruct(sol);
%   Aantal neuronen n:
n = length(mat);
%   Nieuwe 2-vector aanmaken:
dydt = zeros(n, 1);
%   Parameters:
w = mat;
Smax = parm(1);
VT = parm(2);
VS = parm(3);
tau = parm(4);
%   Hier staat de ODE:
for i=1:n
    for j=1:n
        if (i==j)
            dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(y(j)-VT)/VS))));
        else
            if (a==0)
                dydt(i) = dydt(i) + (w(i+(j-1)*n)*( Smax / (1+exp((VT)/VS))));
            else
```

```
                if (t-delay <= (k-2)*delay)
                    ydelay = deval(sol, t-delay+100*eps);
                elseif (t-delay >= (k-1)*delay)
                    ydelay = deval(sol, t-delay-100*eps);
                else
                    ydelay = deval(sol, t-delay);
                end
                dydt(i) = dydt(i) +
                    (w(i+(j-1)*n)*( Smax / (1+exp(-(ydelay(j)-VT)/VS))));
            end
        end
    end
    dydt(i) = (dydt(i) - y(i)) / tau + input(i);
end
```

8.3 Bijlage 3: oplossen met behulp van de ingebouwde tool

```
function dderun = dderun(w, delay, input)
n = length(w);
ydel = (delay * ones(n,1))';
sol = dde23(@ddeneuron,ydel,@ddex1hist,[0, 50], [], w, input);
figure;
plot(sol.x,sol.y)

%-------------------------------------------------------------
function s = ddex1hist(t, w, input)
% Constant history function for DDEX1.
n = length(w);
s = zeros(n,1);

%-------------------------------------------------------------

function dydt = ddeneuron(t, y, Z, w, input)
n = length(w);
Smax = 1;
VT = 4;
VS = 0.1;
tau = 1;
A = zeros(n,1);
%ylag1 = Z(:,1);
%ylag2 = Z(:,2);
for i=1:n
    for j=1:n
        if (i==j)
            A(i) = A(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(y(j)-VT)/VS))));
        else
            A(i) = A(i) + (w(i+(j-1)*n)*( Smax / (1+exp(-(Z(j,j)-VT)/VS))));
        end
    end
    A(i) = (A(i) - y(i)) / tau + input(i);
end

dydt = A;
```

Referenties

[1] B. Gutkin, D. Pinto, B. Ermentrout, (2003) Mathematical neuroscience: from neurons to circuits to systems, Journal of Physiology-Paris, 97, 209-219.

[2] Ch. Fall, E. Marland, J. Wagner, John Tyson, (2002) *Computational Cell Biology*, Springer-Verlag

[3] L. Perko, (2003) *Differential Equations and Dynamical Systems*, Springer-Verlag.

[4] B. Ermentrout, (2003) Neural networks as spatio-temporal pattern-forming systems, Rep. Prog. Phys. 61, 353-430.

[5] V.I. Istrăţescu, (1981) *Fixed point theory; an introduction*, Dordrecht: D. Reidel Publ. Comp.