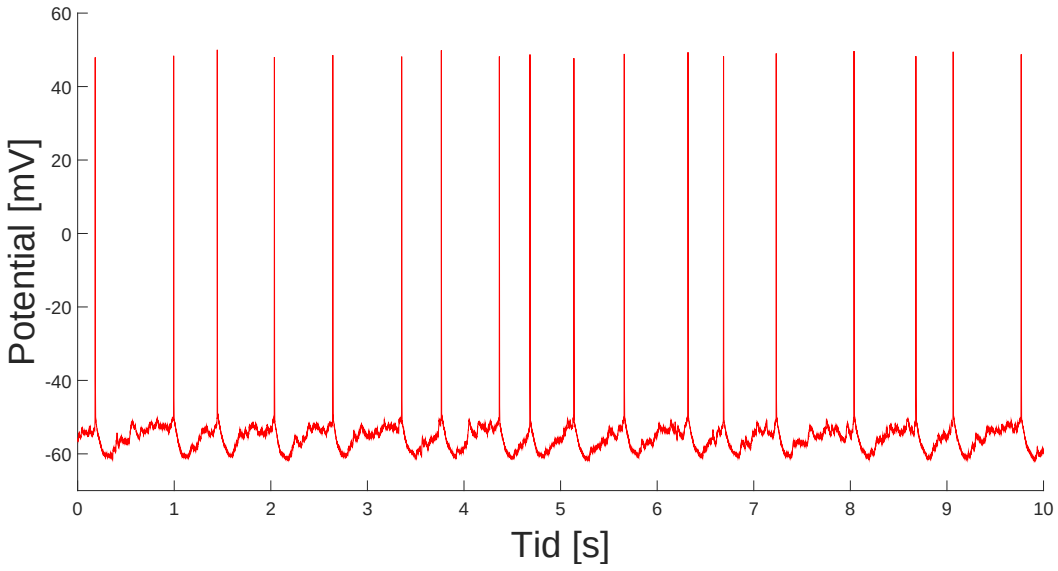
CHALMERS



Modellering av den elektriska potentialen hos enstaka neuroner

En stokastisk modell baserad på mätningar av potentialen hos enstaka pyramidceller i hippocampus

Kandidatarbete inom civilingenjörsprogrammet

Oscar Jonsson, Henrik Norström,
Simon Ovesen och Hilda Sandström

Institutionen för fundamental fysik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2015

Kandidatarbete FUFX02-15-05

Modellering av den elektriska potentialen hos enstaka neuroner

En stokastisk modell baserad på mätningar av potentialen hos enstaka pyramidceller i hippocampus

Oscar Jonsson, Henrik Norström,
Simon Ovesen och Hilda Sandström

CHALMERS

Institutionen för fundamental fysik
Chalmers tekniska högskola
Göteborg 2015

Modellering av den elektriska potentialen hos enstaka neuroner
En stokastisk modell baserad på mätningar av potentialen hos enstaka pyramidceller i hippocampus
Oscar Jonsson, Henrik Norström, Simon Ovesen och Hilda Sandström

Handledare: Måns Henningson
Examinatorer: Daniel Persson och Christoffer Petersson

Kandidatarbete FUFX02-15-05
Institutionen för fundamental fysik
Chalmers tekniska högskola
412 96 Göteborg
Telefon 031 772 1000

Omslag: Simulerad neuronaktivitet i MATLAB utförd med den modell som presenteras i rapporten.

Modellering av den elektriska potentialen hos enstaka neuroner
En stokastisk modell baserad på mätningar av potentialen hos enstaka pyramidceller i hippocampus

Oscar Jonsson, Henrik Norström, Simon Ovesen och Hilda Sandström
Institutionen för fundamental fysik
Chalmers tekniska högskola

Sammandrag

Denna rapport ämnar till att presentera en matematisk modell av den elektriska potentialen hos cellmembranet i en enstaka neuron. Modellen baseras på mätningar på pyramidceller i hippocampus från råttor, gjorda vid Sahlgrenska. Den är inspirerad av den mycket använda Integrate-and-firemodellen. I modellen beror potentialförändringar på slumpmässigt anländande EPSP, spikgenerering och brus. För att konstruera modellen utvecklades metoder för att bestämma parametrar och parameterintervall från mätdata som sedan används i modellen. Resultat från simuleringar presenteras tillsammans med en kvalitativ jämförelse mellan modellen och experimentella värden. Resultatet av jämförelsen var att modellen fångar många men inte alla beteenden som har observerats i mätningarna. Vidare studier på fler mätningar skulle bidra till bättre parameteruppskattningar och kanske bidra till en mer kvantitativ metod för att jämföra modellen med experimentella värden.

Abstract

This report presents an addition to the research currently being made on how to model the activity of the potential of the membrane in a single neuron from a non-biophysical perspective, similar to the Integrate-and-Fire Model. A mathematical model of the electric potential of a single neuron is presented. The model is based on measurements, performed by a reasearch team at Sahlgrenska University Hospital in Gothenburg, of single pyramidal neurons in the in vitro brain slice of the hippocampus. In the model proposed in the report, spikes, randomly arriving EPSPs, and noise account for the change of the membrane potential. In addition to the model, methods for esimating parameters and parameter distributions from the experimental data are descibed. Results from statistic tests and simulations are presented and discussed and along with a qualitative comparison between the model and the experimental data. The comparison shows a satisfactory similarity between the model and the experimental values. However, further studies of a larger set of measurements would help provide a greater knowledge of the distribution of parameters and different behaviours of the neuron.

Nyckelord: neuron, potential, simulering, modell, epsp, poissonprocess, aktionspotential.

Tillkännagivanden

Vi vill tacka Andreas Björefeldt och Eric Hanse för de experimentella mätningarna från Sahlgrenska Universitetssjukhuset som vi har fått ta del av. Vi vill även tacka vår handledare Måns Henningson för all stöttning, vägledning, och för sin entusiasm för projektet.

Oscar Jonsson, Henrik Norström, Simon Ovesen och Hilda Sandström
Göteborg, juni 2015

Innehåll

Figurer

1

Inledning

Hjärnan kallas ofta för forskningens stora utmaning under detta århundrade [1]. EU-komissionens The Human Brain Project är ett exempel på en storskalig satsning på hjärnforskning som initierats det senaste årtiondet. Där samarbetar flera olika discipliner, allt från medicin till neurovetenskap, för att få en bättre förståelse om hjärnan och neurologiska sjukdomar [2]. Ur medicinsk forskningssynpunkt finns det ett ökande intresse då Alzheimers- och Parkinsons sjukdom blir allt större sociala problem för samhället i länder med en åldrande befolkning, då antalet drabbade ökar och även så omvårdnadskostnader [3], [4]. Ur en annan forskningsynpunkt finns intresse av att förstå hjärnan i syfte att kunna skapa intelligenta program genom att efterlikna den. Uppkomsten av intelligenta program är något som många inom forskningsområdet artificiell intelligens förutser kommer hända inom detta århundrade [5].

Studier av hjärnan på cellnivå kan leda till förståelse av hjärnan som helhet. Hjärnan är ett stort nätverk av neuroner som skickar information mellan sig, och som kodar för kroppens funktioner och beteende. En fungerade modell av en neuron skulle kunna vara ett första steg till förståelsen av nätverket den är en del av. Som tidigare nämnt är detta ett mål för olika forskningsområden och vissa framsteg inom forskningen har redan gjorts. Enligt Wang [6] har studier på enstaka neuroner redan hjälpt förståelsen för hur ett visst beteende hos neuronen är kopplat till en viss kroppslig aktivitet eller upplevelse.

Olika modeller som försöker beskriva spänningsförändringar hos neuronen då den mottar och skickar information har utvecklats det senaste århundradet. Vissa modeller försöker beskriva förändringarna utifrån en biofysikalisk modell, främst Hodgkin-Huxleymodellen [7], [8] medan andra mer abstrakta modeller bortser från en del biologiska fenomen och baseras på stokastiska förändringar, ofta versioner av Integrate-and-Firemodellen [7], [9]. De mer abstrakta modellerna har fördelen att de är enklare att modellera än Hodgkin-Huxleymodellen, medan den senare tros ge bäst överensstämmelse med experimentell data [10]. För att få en realistisk modell måste uppskattningar av olika parametrar tas fram från experiment.

1.1 Syfte

I den här rapporten introduceras en matematisk modell som beskriver grundläggande beteenden för den elektriska potentialen hos enstaka neuroner. Modellen har tagits fram utifrån experimentell data och en jämförelse mellan modellen och dessa mätningar presenteras. Även modellens styrkor och brister diskuteras.

2

Bakgrund

Såsom indikerats i inledningen är hjärnan en av de mest komplexa strukturerna som är kända för mänskligheten. Till exempel består den mänskliga hjärnan uppskattningsvis av 10^{12} neuroner, som är sammankopplade i ett stort nätverk. Neuroner är celler som är specialiserade på att motta, behandla och överföra information vidare genom hjärnan och ut till kroppen [11]. En översiktlig beskrivning av hur detta fungerar presenteras nedan. Därefter behandlas de statistiska metoder som är lämpade för att modellera och undersöka vissa processer i neuronen.

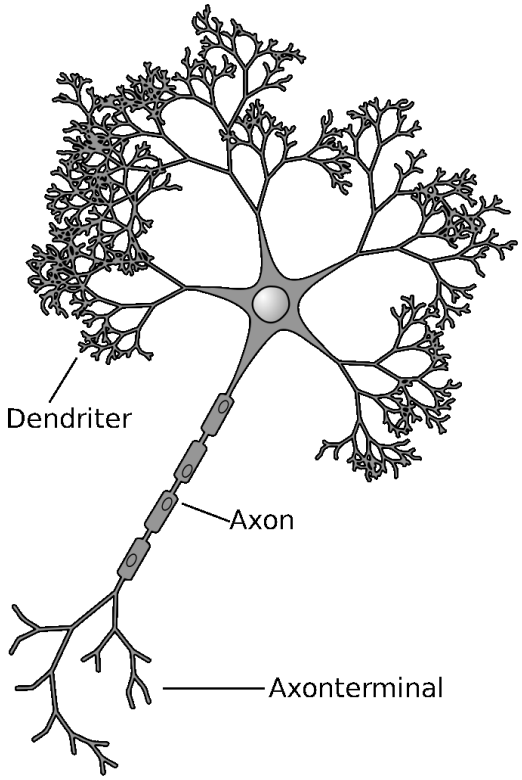
2.1 Kort om hjärnan och neuroner

En neuron består av en cellkropp och ut från cellkroppen sträcker sig olika typer av fiberutskott som möjliggör kommunikation mellan olika neuroner. Axonen är det största utskottet från cellkroppen. Via axonen skickar neuronen ut information och via ett antal mindre utskott, som kallas dendriter, tar cellen emot information. Även cellkroppen kan fungera som direkt mottagare [9]. Platsen för informationsutbytet mellan axonen på en cell till en annan cell kallas för en synaps. Figur 2.1 visar en schematisk bild av en neuron. En enstaka neuron kan motta information via tusentals synapser [11].

Synapser kan vara av två olika typer, kemisk och elektrisk, där den senare är mer ovanlig hos däggdjur. Vid den kemiska synapsen diffunderar transmittorsubstanser från kärl i cellmembranet eller på axonens terminal över ett väldigt litet utrymme mellan axonen och mottagarcellen [11], [13]. Transmittorsubstanserna fäster sedan i receptorer på mottagarcellen vilket öppnar jonkanaler i dess cellmembran, något som i sin tur ändrar potentialen över cellmembranet [11].

Den elektriska potentialen över cellmembranet kan ändras genom att joner tillåts diffundera in och ut ur cellen på grund av koncentrationsgradienter. Cellens vilopotential orsakas av olika koncentration av joner innanför och utanför cellmembranet [11]. Olika värden på vilopotentialen föreslås i litteraturen, från $-65\,\mathrm{mV}$ [9], [14] till mellan -70 och $-80\,\mathrm{mV}$ [11], [15]. De joner som spelar en viktig roll i förändringen av potentialen är främst K^+, Na^+, Ca^{2+}, och Cl^- [11].

Ökningen av den elektriska potentialen hos mottagarcellen, via synapser, kallas för excitatorisk postsynaptisk potential (EPSP) [16]. En EPSP ligger på några enstaka

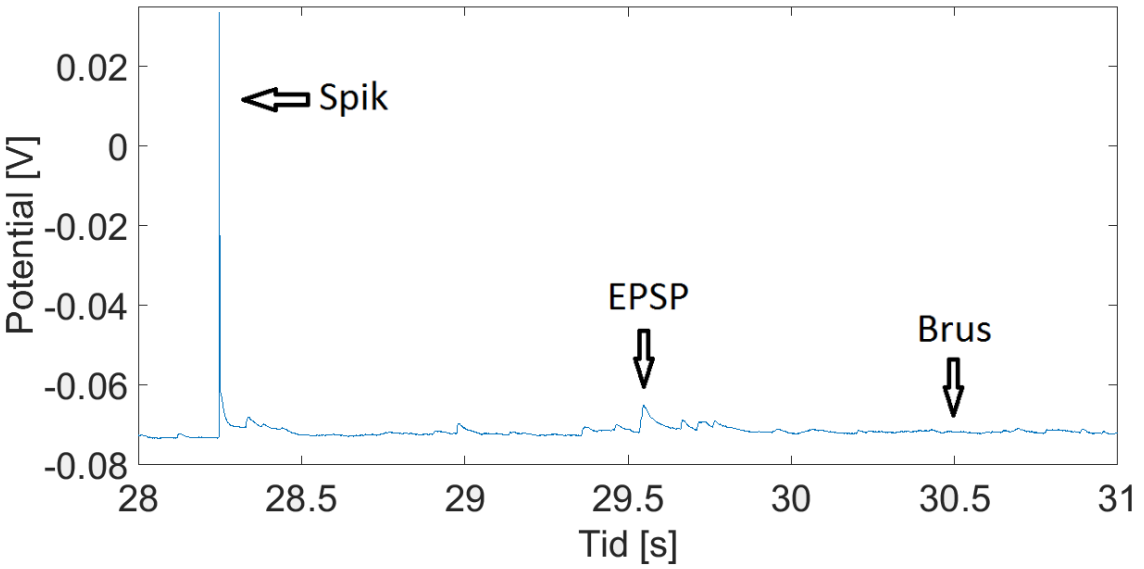


Figur 2.1: En schematisk bild av en neuron. Axonen är det längsta utskottet och via den kan neuronen skicka en signal till andra neuroner. Axonen slutar vid axonterminaler som kopplar till andra neuroner via synapser. Dendriterna syns som de kortare utskotten, och via dem tar neuronen emot signaler. Figuren är en modifierad version av [12] med tillagd text.

mV, och amplituden kan enligt Mason et al. [14] variera mellan 0,05 mV och 2,08 mV. Ur en modelleringssynpunkt är EPSP intressanta då de verkar anlända slumpmässigt i tiden till cellen [17]. Om fler synapser är aktiva samtidigt resulterar detta i större EPSP.

Om ett flertal EPSP ökar potentialen hos cellmembranet på mottagarcellen över ett visst tröskelvärde bildas en aktionspotential, eller spik [11]. Spikens typiska utseende och en tydlig EPSP kan observeras i figur 2.2. Detta tröskelvärde är enligt Fletcher [15] och Mason et al. [14] ungefär 20 mV över vilopotentialen stort, men ett tröskelvärde på 10 mV föreslås istället av [11]. Det här tröskelvärdet är dessutom inte konstant, utan kan variera över tiden [18]. Då tröskelvärdet uppnåtts öppnas spänningskontrollerade natriumkanaler, som normalt är stängda, vilket orsakar den karakteristiska branta ökningen av potentialen då en spik inträffar [15]. Då natriumkanalen är öppen kan inte neuronen motta ny stimuli och generera en ny spik [11]. Kort innebär en spik att en signal skickas från neuronen medan en EPSP uppstår när signaler tas emot.

Spikar har ett väldigt typiskt utseende och varar ofta bara ett par millisekunder [16]. Under en spik ökar potentialen snabbt flera tiotals mV till ungefär -10 till -5 mV



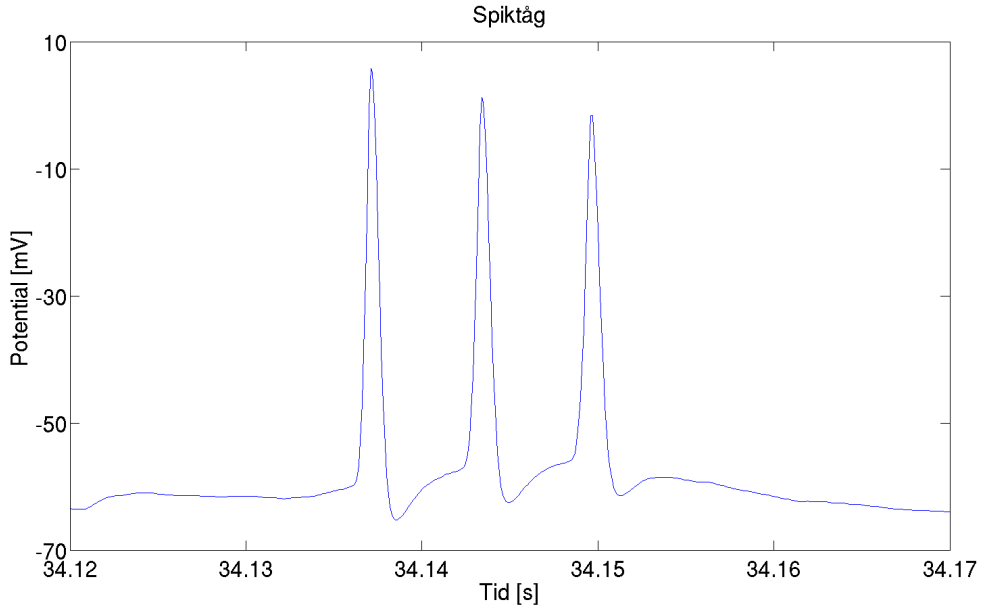
Figur 2.2: Figuren visar olika beteenden hos potentialen. Vid 28,25 s kan en aktionspotential observeras. Potentialen ökar då från ungefär −70 mV till 30 mV. Senare vid ungefär 29,5 s inträffar mindre ökningar och variationer hos potentialen. Detta är ett exempel på EPSP. Vid 30,5 s kan brus observeras som ytterst små variationer hos potentialen.

[11], [16]. Sedan återgår potentialen snabbt till vilopotentialen då kaliumjoner som kan passera membranet fritt rör sig för att motverka ändringen som genererats av passagen av natriumjoner. Dessutom stängs natriumkanalen kort efter att den öppnats, och spänningsstyrda kaliumkanaler öppnas som tillåter ytterligare flöde av kaliumjoner som återställer potentialen [15]. Då potentialen återgår till viloläget sjunker den typiskt under vilopotentialen under en kort period . Detta kallas efterpotential eller efterhyperpolarisation (After hyperpolarization, AHP) [16].

Olika neuroner avfyrar spikar med olika frekvens. Spiktåg är en benämning på utseendet hos potentialen då tidsintervallen mellan spikarna är väldigt litet, enligt Bean [19] mindre än ungefär 5 ms. Spiktåg kan genereras av ett flöde av natriumjoner som fortsätter efter stigningen av potentialen. Detta leder till en efterliggande spänningsökning som i sin tur kan generera en ny spik [19]. Detta beteende kan ses i figur 2.3. Somliga neuroner avfyrar spikar mer eller mindre regelbundet och vissa andra helt oregelbundet, så att ankomsten av spikar i detta fall kan modelleras med en Poissonprocess, (se avsnitt 2.3.3), [20].

Det finns också receptorer som vid aktivering tillåter en minskning av membranpotentialen under vilopotentialen [16]. Om särskilda transmittorsubstanser aktiverar dessa receptorer öppnas kanaler som tillåter klorjoner att passera membranet, varpå en inhibitorisk postsynaptisk potential (IPSP) genereras [11].

Enligt Yarom och Hounsgaard [21] kan potentialen påverkas av jonkanaler som aktiveras tillsammans vilket resulterar i en liten spänningsändring. I de flesta fall kan dock membranets kapacitans agera dämpande för att förhindra snabba ändringar av membranpotentialen. I somliga fall kan dock spänningsändringen förstärkas på



Figur 2.3: Figuren visar ett exempel på ett spiktåg, ett fenomen som uppstår när spikar hamnar väldigt tätt ihop.

grund av positiv återkoppling via en passage av natriumjoner. I synapsen frigörs spontant små mängder transmittorsubstanser, och neuronen tar då in små mängder natriumjoner som ökar potentialen något. Detta resulterar i brus hos potentialen [11]. Dessutom tillkommer brus från yttre källor i mätningarna, såsom mätutrustningen. Bruset kan också observeras i figur 2.2.

Den vätska som hjärnan, och därmed även neuronerna, befinner sig i kallas cerebrospinalvätska, eller CSF (efter cerebrospinal fluid). Denna vätska innehåller bland annat joner och diverse organiska molekyler. I experimentella sammanhang används dock ofta en artificiell ersättning som innehåller samma jonkoncentrationer, men saknar det organiska innehållet. För att skilja dessa åt används beteckningarna aCSF för den artificiella och hCSF för den mänskliga [22].

Hippocampus är en hjärnstruktur som är inblandad i processer kopplade till minnet. Den är av intresse för studier då många neurologiska sjukdomar är kopplade just till hippocampus[16]. Pyramidneuroner återfinns i hippocampus och kallas så som de gör på grund av att deras cellkropp är triangelliknande till formen. Det har upptäckts att de har koppling både till Alzheimers sjukdom och epilepsi, och att de således är inblandade i kognitiva processer[23].

2.2 Integrate-and-fire-modellen

Idén som nämnts i föregående avsnitt, om att en neuron avfyrar en signal då dess membranpotential nått sitt tröskelvärde utnyttjas i en av de mest använda modelleringsmetoderna som kallas Integrate-and-fire-modellen, IFM, som Burkitt också

sammanfattar i [17]. I IFM modelleras neuronen som en parallel RC-krets för att redogöra för uppladdningen av joner på en sida av cellmembranet, vilket tänks motsvara uppladdningen av en kondensator. Då spänningen över kondensatorn nått till sitt tröskelvärde laddas den ur genom att tillåta en ström passera [9], [17].

Uppladdningen av kondensatorn från synaptiska bidrag har i olika undersökningar av metoden modellerats på olika sätt men består i regel av tidsstokastisk indata, vilket motsvarar signaler som mottas av cellmembranet eller dendriterna hos neuronen. Mest vanligt modelleras ankomsten av indatat stokastiskt via en Poissonprocess, se avsnitt 2.3.3, då tidsintervallen mellan dem antas oberoende [17].

IFM representeras av förändring i spänningen över cellmembranet. Storheter som påverkar spänningen är ström från synapser (signaler från andra neuroner), ström från läckage genom cellmembranet. Även pålagd ström, från en elektrod i neuronen, kan inkluderas som ett bidrag. Vidare antas det att cellmembranet har en konstant kapacitans och resistans för passivt läckage ur cellen [17]. Termen som beskriver läckaget, eller med andra ord en spontan minskning av membranpotentialen har introducerats i senare versioner av IFM [9].

Indata från synapser representeras som viktade Dirac-funktioner som anländer vid olika tidpunkter. Bidrag från alla olika synapser summeras för att ge det totala bidraget av ström från synapser. Den resulterande tidsberoende svarsfunktionen från en synaps beror på förhållandet mellan cellmembranets passiva tidskonstant (definierad som produkten av cellmembranets kapacitans och resistans), och en synaptisk tidsfaktor [17]. Detta tas upp senare i 4.2.3.

2.3 Relevanta statistiska begrepp

Som tidigare nämnts kan ankomsten av EPSP verka slumpmässig. För att kunna modellera uppkomsten av både spikar och EPSP behöver olika stokastiska modeller prövas för att se om de verkar producera likartade fördelningar. Olika metoder för att anpassa parametrar och undersöka hur bra modellen överensstämmer med experimentella värden behöver också användas.

2.3.1 Stokastiska variabler och sannolikhetsfördelningar

En grundpelare vid stokastisk modellering är stokastiska variabler. En stokastisk variabel X är en variabel som definieras av en sannolikhetsfördelning i stället för faktiska tal. Detta innebär att vid en observation x av variabeln så finns det en viss sannolikhet att variabeln har ett visst värde. För diskreta stokastiska variabler, alltså de där det endast finns ett disktret antal utfall, beskrivs denna sannolikhet med en sannolikhetsfunktion eller pmf (efter probability mass function) $p(x)$ [24] som definieras enligt

$$p(x) = P(X = x), \quad p(x) \geq 0 \tag{2.1}$$

där P betecknar sannolikhet. En relaterad funktion är den kumulativa fördelningsfunktionen eller cdf (efter cumulative distribution function) $F(x)$, som definieras enligt

$$F(x) = P(X \leq x) = \sum_{k=1}^{x} p(k), \quad F(x) \leq 1. \tag{2.2}$$

Ett exempel på en diskret sannolikhetsfördelning är Poissonfördelningen. Sannolikhetsfunktionen för en Poissonfördelad stokastisk variabel har formen

$$p(x;\lambda) = \frac{(\lambda)^x}{x!}e^{-\lambda}, \tag{2.3}$$

där λ är en parameter som bestämmer fördelningens form. Både väntevärde och varians beskrivs av denna parameter ($\mathrm{E}[X] = \mathrm{Var}[X] = \lambda$) [24].

Stokastiska variabler med utfall definierade på ett reellt intervall kallas kontinuerliga stokastiska variabler. Dessa beskrivs av en täthetsfunktion eller pdf (efter probability density function) $f(x)$ som definieras enligt [24]

$$P(a < X < b) = \int_a^b f(x)\,\mathrm{d}x, \quad f(x) \geq 0 \tag{2.4}$$

samt en cdf

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)\,\mathrm{d}x, \quad F(x) \leq 1. \tag{2.5}$$

En annan intressant funktion som kan associeras till stokastiska variabler är kvantilfunktionen. Denna är definierad som inversen till den kumulativa fördelningsfunktionen, $Q(x) = F^{-1}(x)$. $Q(\frac{1}{q})$ kan tolkas som det x för vilket $\frac{1}{q}$ av alla tänkbara utfall under x har passerats. Dessa värden kallas kvantiler (specifikt q-kvantiler), där den förmodligen mest välkända är 2-kvantilen, eller medianen [24].

Exempel på kontinuerliga sannolikhetsfördelningar är exponentialfördelningen

$$f(x;\lambda) = \lambda e^{-\lambda x},\ x \in [0,\infty),\ \lambda > 0 \tag{2.6}$$

$$F(x;\lambda) = 1 - e^{-\lambda x},\ x \in [0,\infty),\ \lambda > 0 \tag{2.7}$$

$$\mathrm{E}[X] = \frac{1}{\lambda}, \quad \mathrm{Var}[X] = \frac{1}{\lambda^2}, \tag{2.8}$$

och gammafördelningen,

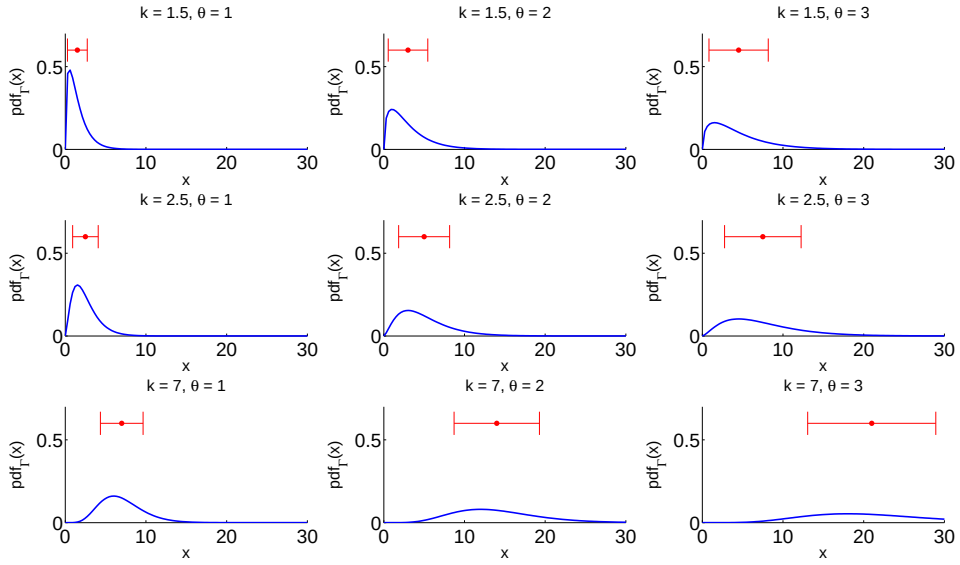
$$f(x;k,\theta) = \frac{1}{\Gamma(k)\theta^k}x^{k-1}e^{-\frac{x}{\theta}},\ x,k,\theta > 0 \tag{2.9}$$

$$F(x;k,\theta) = \frac{1}{\Gamma(k)}\gamma\left(k,\frac{x}{\theta}\right),\ x,k,\theta > 0 \tag{2.10}$$

$$\mathrm{E}[X] = k\theta, \quad \mathrm{Var}[X] = k\theta^2, \tag{2.11}$$

där gammafunktionen $\Gamma(x)$ och den undre inkompletta gammafunktionen $\gamma(x)$ beskrivs enligt

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,\mathrm{d}t, \quad \gamma(s,x) = \int_0^x t^{s-1}e^{-t}\,\mathrm{d}t. \tag{2.12}$$



Figur 2.4: Denna figur illustrerar hur gammafördelningens egenskaper beror på dess parametrar k och θ. De blå kurvorna representerar fördelningens pdf medan de röda linjerna beskriver ett två standardavvikelser brett intervall centrerat kring väntevärdet.

Det kan lätt ses att exponentialfunktionen med parameter $\frac{1}{\theta}$ är ett specialfall av gammafunktionen med parameter $k = 1$ [25].

Det är lätt att se att ett högre λ innebär en snabbare avtagande exponentialfunktion i exponentialfördelninen. Gammafördelningens utseende är dock inte lika trivialt. Hur denna fördelning beror på k och θ kan därför ses i figur 2.4.

2.3.2 Stokastiska processer

Det finns många förlopp som beror av så pass många variabler att det är enklast att modellera dem med sannolikhetsbaserade metoder. Ett exempel på ett sådant förlopp är uppkomsten av EPSP som här kommer att modelleras som en stokastisk process.

En stokastisk process är en samling stokastiska variabler definierade på ett sannolikhetsrum. Matematiskt skrivs en stokastisk process $\{X(t); t \in T\}, T \subset \mathbb{R}, X(t) \in \mathbb{R}^n$ [25]. Till skillnad från en deterministisk process som alltid ger samma resultat för samma indata (exempelvis en differentialekvation eller en funktion) så är $X(t)$ en stokastisk variabel för fixt t. Stokastiska processer används framför allt för att modellera processer som kan tyckas slumpmässiga, även om de sällan är det i verkligheten.

En typ av stokastisk process är en så kallad räkneprocess $\{N(t), t \geq 0\}$. Denna har egenskaperna

- $N(t) \in \mathbb{N} \cup \{0\}$
- $s \leq t \Rightarrow N(s) \leq N(t)$,

och används för att räkna antalet händelser som inträffat innan tiden t.

2.3.3 Poissonprocessen

Ett exempel på en sådan process är Poissonprocessen, en räkneprocess med följande egenskaper:

- $P(N(0) = 0) = 1$.
- Händelserna som räknas är oberoende av varandra.
- $N(t)$ är Poissonfördelad med parameter λt, där λ är den genomsnittliga händelsefrekvensen.
- Poissonprocessen saknar minne, vilket innebär att $N(\tau)$ och $N(t + \tau) - N(t)$ har samma sannolikhetsfördelning.
- Två händelser kan inte inträffa samtidigt.

En konsekvens av dessa egenskaper är att tidsintervallen mellan två händelser är exponentialfördelade med parameter λ [24].

Denna typ av Poissonprocess, där λ är konstant, kallas homogen. Homogena Poissonprocesser används ofta för att modellera system där händelser kan antas vara oberoende av varandra och takten händelser sker i inte ändras, exempelvis inkommande samtal i en telefonväxel [26].

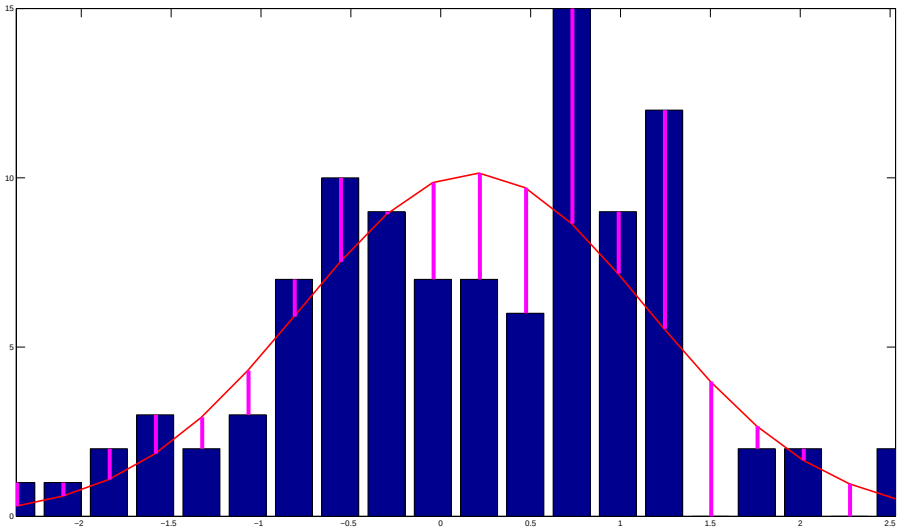
2.3.4 Statistiska tester

När stokastiska modeller anpassas efter mätpunker är det vanligt att se om dessa kan beskrivas med hjälp av sannolikhetsfördelningar. Detta går till genom att välja en lämplig sannolikhetsfördelning och tillämpa en anpassningsmetod. Det finns dock ingen garanti för att valet av fördelning var bra, även om metoden använts korrekt. För att detta finns det ett antal tester som kan genomföras.

Ett exempel på ett sådant är χ^2-testet. Testet går till genom att dela in datan i n intervall och undersöka statistikan

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}, \tag{2.13}$$

där O_i och E_i är det observerade respektive förväntade antalet datapunkter innanför intervall i. Detta kan ses illustrerat i figur 2.5. Själva statistikan konvergerar mot en χ^2-fördelning med parameter $n - k$, där k är antalet frihetsgrader som ska dras



Figur 2.5: Denna figur illustrerar hur χ^2-testets statistika tas fram. Histogrammet består av genererade normalfördelade variabler med en störning bestående av vitt brus pålagd. Den blå kurvan beskriver en anpassad normalfördelning, och de magentafärgade vertikala linjerna motsvarar avståndet mellan det observerade antalet mätpunkter mot det teoretiskt förväntade inom varje intervall. χ^2-statistikan består då av summan av kvadraterna på dessa linjers längder.

bort [27]. χ_2-fördelningen kan beskrivas som en gammafördelning med parametrar $k \to \frac{k}{2}$ och $\theta \to 2$.

Därefter beräknas det så kallade p-värdet, vilket motsvarar sannolikheten att en observation av en slumpvariabel X med denna fördelning är större än statistikan,

$$p = P(X > \chi^2). \tag{2.14}$$

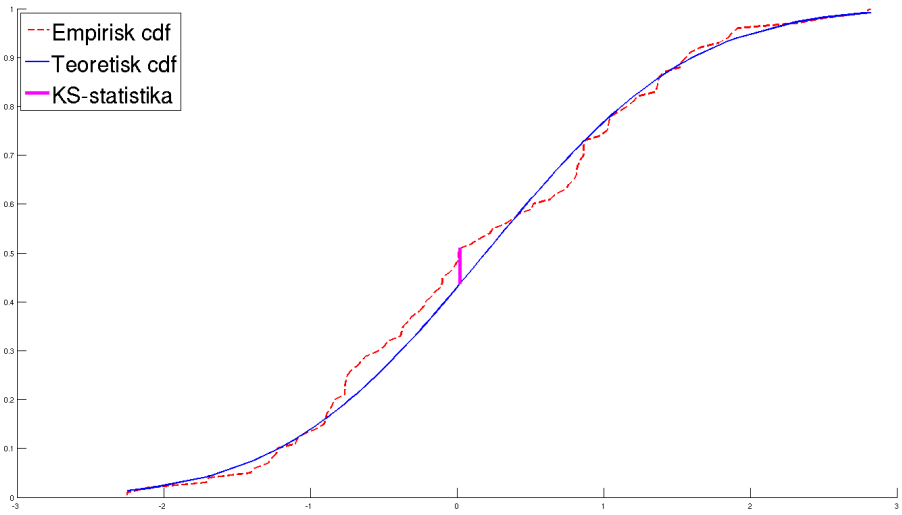
Hypotesen att datan kommer från den föreslagna modellen måste i så fall förkastas under signifikansnivå α om $p < 1 - \alpha$, där ett typiskt värde på α är 0.95 [25].

Detta test fungerar utmärkt för diskreta fördelningar eftersom antalet intervall (eller i detta fall kategorier) enkelt kan väljas till antalet möjliga utfall som observerats. För kontinuerliga fördelningar uppstår däremot problemet att antalet intervall kan väljas rätt godtyckligt, och att resultatet kan bero på detta val. Det kan dock visas att ett bra värde på detta antal är en femtedel av antalet mätpunkter [27], [28].

Ett annat test som finns är Kolmogorov-Smirnovtestet (KS-testet). Den relevanta statistikan D_N i detta test ges av

$$D_N = \sup \left| \hat{F}(x) - F(x) \right|, \tag{2.15}$$

där $\hat{F}(x)$ och $F(x)$ är empirisk respektive teoretisk cdf och N är antalet mätpunkter. Detta värde kan ses illustrerat i figur 2.6. Det går att visa att $\sqrt{N}D_N$ konvergerar



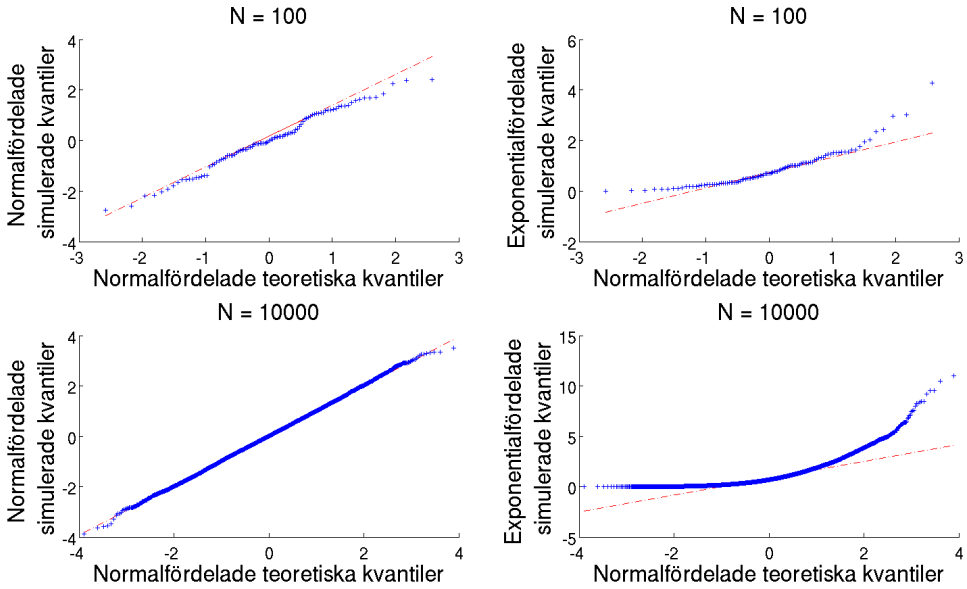
Figur 2.6: Denna figur illustrerar KS-statistikan. Den röda streckade kurvan beskriver en empirisk cdf för genererade normalfördelade variabler med en störning bestående av vitt brus. Den blå kurvan beskriver motsvarande teoretiska cdf, och den tjocka linjen representerar det största avståndet mellan dessa två. Avståndet på denna linje utgör KS-statistikan D_N.

mot en Kolmogorovfördelning (definieras inte i denna rapport). Denna jämförs sedan med p-värdet på samma sätt som innan [25].

KS-testet har ett antal fördelar, men det har också en begränsning som är kraftigt hämmande för denna undersöknings syfte. Testet gäller endast när den teoretiska fördelningen, inklusive parametrar, är känd sedan innan. Om en metodik kräver att parametrarna uppskattas utifrån datan kan alltså testet inte tillämpas korrekt eftersom den teoretiska kumulativa sannolikhetsfunktionen kommer bli beroende av den empiriska [29].

Det finns dock i vissa fall en väg runt detta problem. Hubert Lilliefors har utvecklat det så kallade Lillieforstestet som använder samma tillvägagångssätt som KS-testet, men jämför med Lillieforsfördelningen i stället för Kolmogorovfördelningen. Denna fördelning har kompenserat för felet som uppstår av att parametrarna är uppskattade från datan. Tabellvärden har tagits fram genom simuleringsmetoder, men dessa tabeller täcker i nuläget endast normal- och exponentialfördelningarna [29].

En annan metod som kan användas för att ge en kvalitativ bild av en anpassnings kvalitet är så kallade Q-Q-plottar (där Q står för Quantile, eller kvantil). Denna metod går till genom att rita upp två fördelningars kvantiler mot varandra. Om båda fördelningar består av empirisk data används de empiriska kvantilerna. Detta görs enklast om de har samma antal mätpunkter, eftersom allt som då behövs göras är att använda mätpunkterna sorterade i storleksordning. Om en mängd empirisk data i stället ska jämföras med en teoretisk fördelning används i stället den teoretiska kvantilfunktionens motsvarande värden [24].



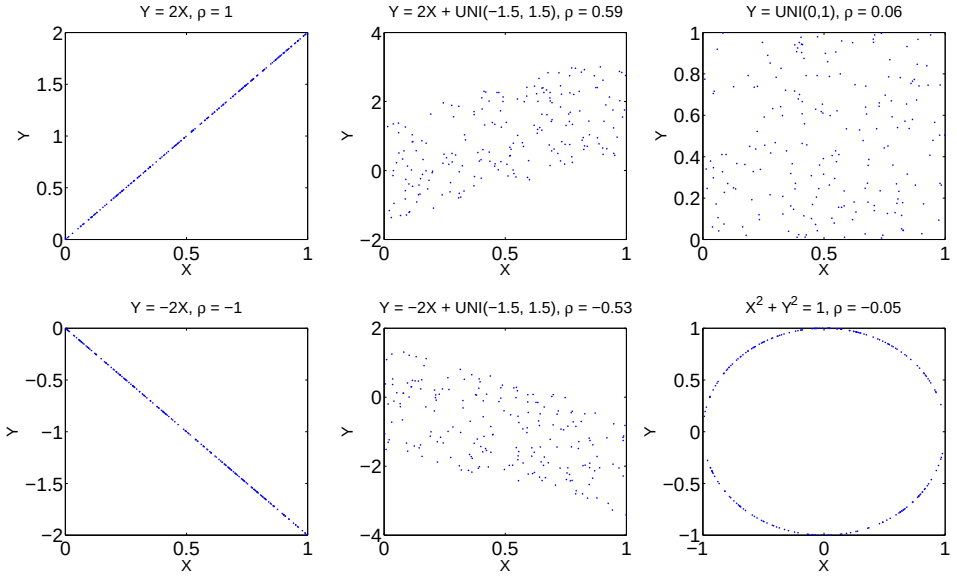
Figur 2.7: Figuren visar exempel på olika Q-Q-plottar med N antal punkter. Figurens syfte är att illustrera de linjära respektive icke linjära egenskaperna hos Q-Q-plottar beroende på om datamängderna kommer från samma fördelning eller inte, samt hur antalet punkter kan påverka grafens kvalitet. De två vänstra visar kvantilerna för genererade normalfördelade slumpvariabler mot de från den teoretiska kvantilfunktionen. I de två högra är de genererade variablerna i stället exponentialfördelade.

Om de två datamängderna kommer från samma fördelning bör Q-Q-plotten bilda en relativt rät linje. Om den i stället visar ett kraftigt icke linjärt beteende så hör de sannolikt till olika fördelningar [24]. Figur 2.7 illustrerar denna skillnad genom att rita upp en graf för genererade normalfördelade slumpvariabler mot en teoretisk normalfördelning, och en där det i stället är exponentialfördelade variabler som genererats. Figuren visar även hur grafens kvalitet kan beror på hur många mätpunkter som används.

Ytterligare ett mått som kan vara användbart är Pearsons produktmomentkorrelationskoefficient (hädanefter bara kallad korrelationskoefficient). Den används för att undersöka eventuell korrelation mellan två datamängder X och Y. Om dessa har standardavvikelserna σ_X och σ_Y så definieras korrelationskoefficienten ρ som

$$\rho = \frac{\mathrm{COV}[X, Y]}{\sigma_X \sigma_Y} \Rightarrow \rho \in [-1, 1]. \tag{2.16}$$

Att $\rho = 0$ innebär avsaknad av korrelation, $\rho = 1$ innebär total korrelation och $\rho = -1$ total antikorrelation. Korrelationskoefficienten kan däremot endast hitta linjära beroenden; den märker inte av icke linjära alls [24]. Exempel på dessa beteenden kan ses i figur 2.8.



Figur 2.8: Denna figur illustrerar beteende hos Pearsons produktmomentkorrelationskoefficient ρ. För rent linjära korrelationer mellan variablerna X och Y ger koefficienten ett värde nära ± 1, vilket kan ses i de vänstra graferna. För samband som endast delvis innehåller linjära komponenter fås ett till beloppet mindre värde, vilket kan ses i mittengraferna. I de högra kan ses att total brist på linjärt beroende (även om icke linjär korrelation finns) ger ett värde nära 0. UNI(a,b) representerar likformigt fördelade slumpvariabler mellan a och b.

Slutligen finns det ett antal mått för då datan har anpassats efter en deterministisk modell genom exempelvis minsta kvadratmetoden. Två av dessa, SSE och SST, definieras enligt

$$\text{SSE} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.17}$$

$$\text{SST} = \sum_{i=1}^{N} (y_i - \bar{y})^2, \tag{2.18}$$

där N är antalet datapunkter, y är den uppmätta datan, $\bar{y}$ är dennas medelvärde och $\hat{y}$ är den anpassade modellen [30]. Dessa mått har sina användningsområden, men en möjlig nackdel är att de är absoluta och därför inte säger så mycket utan kunskap om storleken på y och $\hat{y}$. Av denna anledning används även ett relativt mått kallat R^2, vilket definieras som

$$R^2 = 1 - \frac{SSE}{SST}. \tag{2.19}$$

Den andra termen kan i detta mått tolkas som ett mått på hur stor del av den totala variansen som är oförklarad av modellen. Med andra ord innebär ett stort (nära 1) värde på R^2 att modellen förklarar en stor del av den totala variansen, medan ett litet (nära 0) innebär motsatsen [24].

R^2 har däremot en inbyggd bias, vilket innebär att den oftast visar ett värde som är något högre än det borde vara. För att undgå detta kan justerad (adjusted) R^2 användas. Detta definieras som

$$R^2_{adj} = 1 - (1 - R^2)\frac{N-1}{N-p} = 1 - \frac{N-1}{N-p}\frac{SSE}{SST}, \tag{2.20}$$

där p är antalet parametrar som modellen anpassats efter. R^2_{adj} är utan bias, och visar därför ofta ett något mindre värde än R^2 [31].

3

Metod

Den analyserade datan kommer från Sahlgrenska universitetssjukhuset, där den ursprungligen användes för en studie som jämförde hCSF och aCSF. I mätserien gjordes 18 mätningar av potentialen på totalt 13 olika neuroner. Neuronerna i fråga var pyramidceller i hippocampus. Vid mätningarna var neuronerna nedsänkta i antingen hCSF eller aCSF. Med ett tidsintervall på 60 s och en samplingsfrekvens på 10 kHz gick det att observera utseende och beteende på spikar och EPSP, både kvalitativt och kvantitativt. Med kvalitativt menas en analys som fokuserar på mönster och trender i det som analyseras. En kvantitativ analys kan istället presenteras med siffror, från till exempel statistiska test. Databehandling skedde med hjälp av MATLAB.

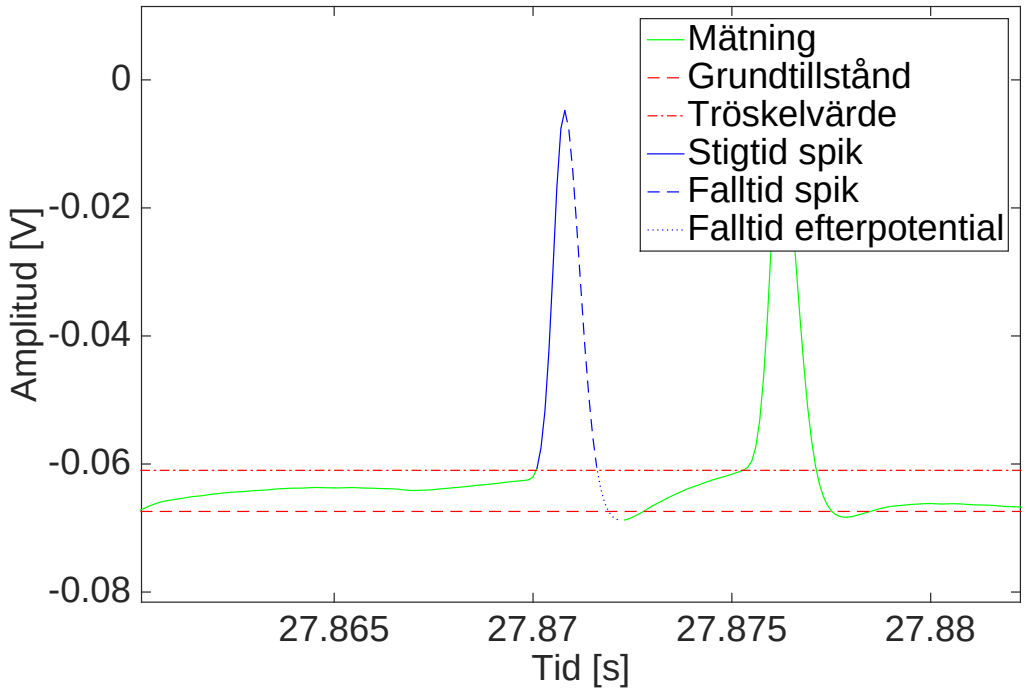
För att få mer struktur på projektet delades modellen in i flera mindre modeller som beskriver olika beteenden: spikar, EPSP och brus. Dessa tre komponenter har var för sig återkommande egenskaper och blev därför lättare att hantera separerat. I figur 2.2 kan den här uppdelningen observeras. Att främst behandla olika fenomen som kunde observeras i mätningarna är en avgränsning för modellen som gjorts. Nedan presenteras de olika metoder som använts för att undersöka mätningarna. Dessa metoder är de som har visat sig fungera bäst utav de olika metoder som har provats under arbetets gång.

3.1 Spikens storheter och egenskaper

För att analysera spikens beteende behövde den delas upp i mindre delar att undersöka. Tröskelvärdet definierades som i avnsitt 2.1 och användes för att definiera stig- och falltid för spikarna. Det förklaras även hur amplituden för spikarna och efterpotentialen hanterades.

3.1.1 Tröskelvärde, stigtid och falltid

Tröskelvärdet för varje enskild mätning definierades som den högsta potentialen som inte utlöst en spik. Genom tröskelvärdets definition kunde startpunkten för en spik definieras. På samma sätt definierades en sluttid när spiken passerade ner under tröskelvärdet. Genom att utgå från maxpunkten definierades både stigtid och



Figur 3.1: Figuren visar en mätning tillsammans med definierat tröskelvärde och grundtillstånd. För den vänstra spiken är stigtid, falltd och falltid för efterpotentialen markerade.

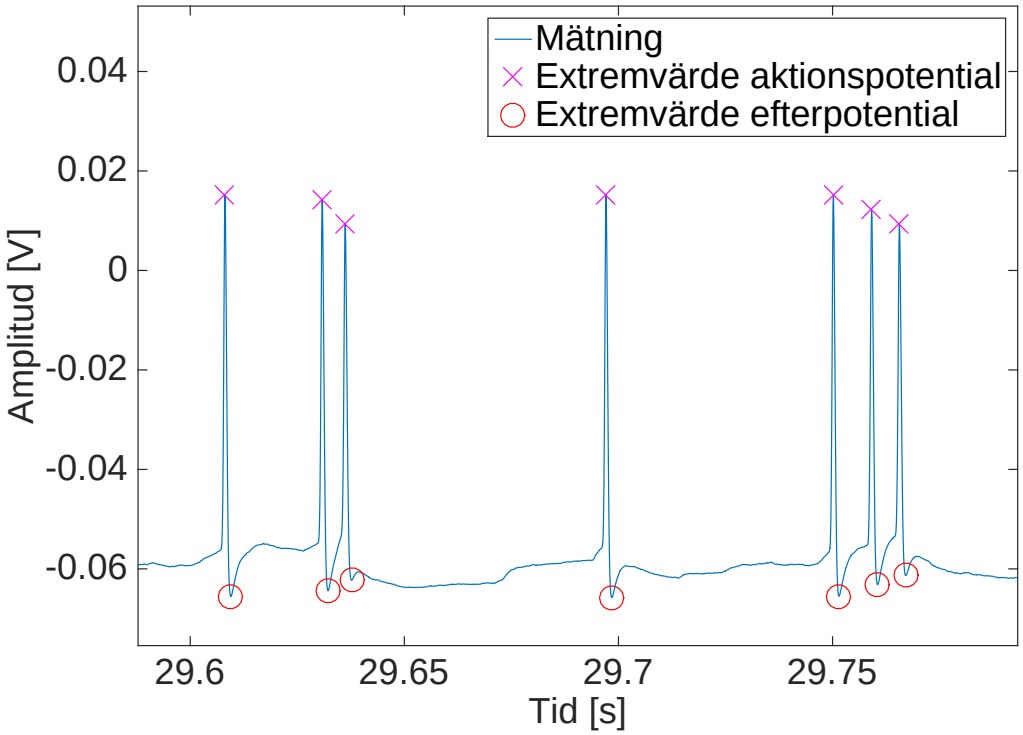
falltid. På liknande sätt definierades falltiden för efterpotentialen som tidsavståndet mellan tröskelvärdet och efterpotentialens lägsta punkt. En grundnivå beräknades genom att beräkna medelvärdet av punkter som med ögonmått bedömdes att ligga nära grundnivån. Samtliga definitioner går att observera i figur 3.1.

Ett mer statistiskt bestämt mått på grundnivån var svårt att få. Eftersom den sökta nivån ska vara ett lägstavärde som potentialen sällan går under så skulle ett medelvärde, även om spikar tas bort, ge ett alldeles för högt värde. Att definiera det lägsta värdet som grundnivå var inte heller möjligt, eftersom många efterpotentialer kunde vara märktbart lägre än grundnivån.

3.1.2 Amplitud

Maximum för spikar hos mätdatan lagrades som den högsta punkten varje gång potentialen gjort två passager förbi tröskelvärdet, alltså passerat över och sedan tillbaka under. Sedan definierades amplituden som avståndet mellan tröskelvärdet och maxpunkterna. För varje mätning kunde sedan ett medelvärde för spikamplituden enkelt definieras. I figur 3.2 syns hur spikarnas maxvärde registreras.

Undersökningen av hur spikarnas amplitudvariation skulle modelleras gjordes genom att titta på hur amplitudskillnaden berodde på motsvarande tidsintervall mellan två närmsta grannar, samt det procentuella avståndet till medelamplituden. Med hjälp



Figur 3.2: Figuren visar hur vår algoritm placerar ut extremvärden för spikar och efterpotentialer. Vid 29,65 sekunder kan vi se att programmet inte registrerat potentialen som ett extremvärde trots att det är lägre än efterpotentialen strax tidigare.

av verktyget *cftool* (curve-fitting tool) i MATLAB kunde en funktion anpassas med tidsintervallet och det procentuella avståndet till medelamplituden som inparametrar.

3.1.3 Efterpotential

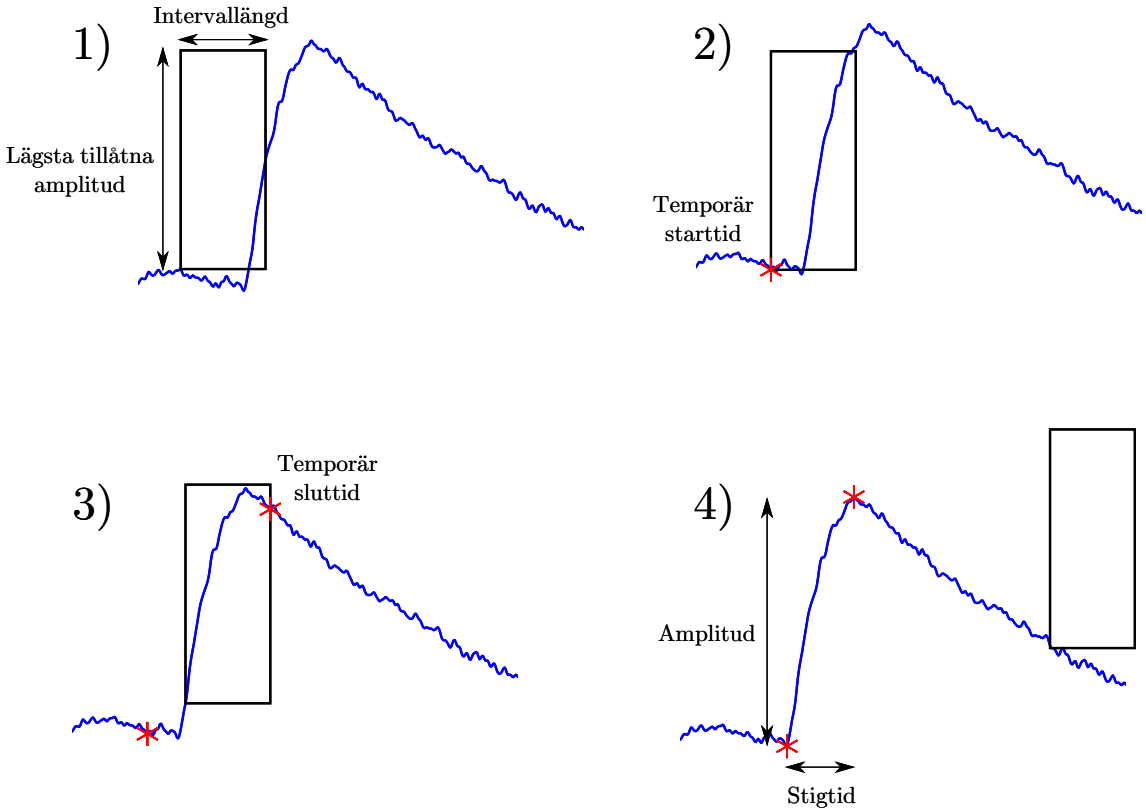
För mätningar där efterpotentialen var lägre än grundnivån lagrades den lägsta punkten mellan två spikar. Amplituden för efterpotentialen definierades sedan som det positiva avståndet mellan tröskelvärdet och minimum. För mätningar där efterpotentialen inte var den lägsta punkten mellan två spikar fick datahämtningen istället utföras genom att hitta minimum inom ett givet avstånd efter varje spik. Detta fick anpassas för varje enskild mätning. I figur 3.2 syns hur extremvärdet för några efterpotentialer är registrerade för en mätning.

3.2 Identifiering av EPSP

Det var svårt att ur mätningarna på ett konsekvent sätt urskilja vad som var en EPSP. Vissa var så små att de inte kunde särskiljas från brus, och många var super-

ponerade på varandra vilket kraftigt försvårade mätande av falltid. Det kunde dock tydligt ses att en EPSP hade betydligt kortare stigtid än falltid. Av denna anledning valdes delen då potentialen stiger från EPSP:ns startpunkt till dess maxpunkt som det som användes för att karaktärisera en EPSP.

Algoritmen som användes för leta efter EPSP, gick till genom att svepa ett intervall av en fix längd i tidsled en mätpunkt i taget. För varje iteration undersöktes om skillnaden mellan intervallets högsta och lägsta potentialvärde var tillräckligt stor. Om den var det sparades intervallets startpunkt som en temporär starttid för EPSP:n. Intervallet fortsatte därefter att svepas längs tidsaxeln tills kravet på potentialskillnad inte längre kunde uppfyllas. EPSP:ns startsegment definierades då som intervallet mellan hela intervallets min- och maxpunkt. För att reducera antalet falska utslag klipptes dessutom EPSP med för låg amplitud eller för låg lutning mellan start- och slutpunkterna bort. Algoritmen kan ses illustrerad i figur 3.3.



Figur 3.3: Figur som illustrerar den algoritm som användes för att identifiera EPSP från datan. Ett tidsintervall av konstant längd sveps över datan en mätpunkt i taget. För varje mätpunkt undersöks om skillnaden mellan intervallets minsta punkt och dess största är tillräckligt stor (något som ej uppfylls i 1)). I 2) har detta krav uppfyllts, och en temporär starttid markeras i intervallets början. Intervallets svepning fortsätter sedan tills kravet inte längre uppfylls, varpå ett temporärt slutindex markeras vid intervallets slut (3)). Slutligen bestäms slutgiltiga start- och sluttider som min- respektive maxpunkterna inom intervallet mellan de temporära start- och sluttiderna. EPSP:ns stigtid och amplitud definieras så som skillnaden mellan dessa punkter i tids- respektive spänningsled.

De parametrar som styr algoritmen är alltså intervallets längd, minsta amplitudskillnad i ett intervall, minimigränsen för en EPSP:s amplitud och minimigränsen för den totala lutningen. Dessa bestämdes genom ögonmått och fick varieras mellan olika mätningar.

De egenskaper som undersöktes var stigtid och tidsavstånd mellan olika EPSP. Stigtiderna anpassades efter en gammafördelning och tidsavstånden efter en exponentialfördelning. Gammmafördelningarna motiverades av rena utseendeskäl och exponentialfördelningen valdes eftersom det var intressant att undersöka om generering av EPSP kunde modelleras som en Poissonprocess.

Anpassningen till sannolikhetsfördelningar gjordes med en metod som kallas Maximum Likelihood (se bilaga A för detaljer). Hur väl anpassningarna motsvarade datan undersöktes med hjälp av ett par test. Dels användes χ^2-testet (och även Lilliefors-testet för tidsintervallen). Dessutom användes Q-Q-plottar för att ge en rent grafisk bild.

Även amplituden kunde anpassas efter en gammafördelning. Det visade sig dock att stigtid och amplitud hade en icke-trivial korrelationskoefficient. Eftersom oberoende generering av stigtid och amplitud skulle ignorera detta samband behövde någon form av beroende fastställas.

Den modell som användes var att ansätta en stokastisk process

$$A(t_r) = at_r + b + \tilde{X}, \tag{3.1}$$

där A är amplitud, t_r är stigtid, $\tilde{X}$ är en stokastisk variabel och a och b är konstanter. Genom att undersöka fördelningen av

$$A_{\text{uppmätt}} - (at_{r_{\text{uppmätt}}} + b), \tag{3.2}$$

där a och b togs fram med minsta kvadratmedoden, så kunde fördelningen av $\tilde{X}$ undersökas. Det visade sig att även denna formmässigt såg ut att kunna anpassas av en gammafördelning, men detta förhindrades av att $A_{\text{uppmätt}} - (at_{r_{\text{uppmätt}}} + b)$ antog negativa värden. I stället infördes den stokastiska variabeln

$$X = \tilde{X} - c, \tag{3.3}$$

där konstanten c translaterar $\tilde{X}$ så att $X > 0$. Denna nya variabel anpassades efter en gammafördelning och anpassningens kvalitet testades med samma metoder som använts tidigare.

3.3 Brusets frekvensegenskaper

Bruset undersöktes genom att först filtrera bort spikarna och EPSP från mätdatan, genom att skära bort stora potentialer. Den återstående datan genomgick snabb Fouriertransform (fft) för att få fram frekvensspektrumet för bruset. Frekvenstopparna från de olika mätningarna antecknades och jämfördes för att se om det fanns

några gemensamma frekvenser. Bruset framställdes genom en summa av sinustermer innehållande de gemensamma frekvenserna. Amplituden för var och en av termerna anpassades så att de liknade mätningarna så mycket som möjligt. Efter detta infördes en liten slumpmässig störning på bruset för att få en liten oregelbundenhet och för att även blanda in lite av andra frekvenser. Till sist skalades brusets amplitudkomponenter ner till en storlek på µV.

3.4 Avgränsningar

Detta avsnitt behandlar de avgränsningar som val av modell och metoder resulterade i. För det första så är modellen empirisk och baserad på de 18 mätningarna under 60 s som gjordes vid en studie på Sahlgrenska universitetssjukhuset. Dessa mätningar utgör inte en stor mängd data. Dessutom var undersökningen av datan rent fenomenbaserad. Det var utifrån de tre beteendena spik, EPSP och brus som den experimentella datan undersöktes. Den experimentella datan togs fram för att undersöka skillnaden på beteendet hos potentialen då neuronen omgavs av aCSF och hCSF. I undersökningarna av mätningarna som gjorts i syfte för modellen gjordes dock ingen särskiljning mellan de båda.

Valet av metoder för att undersöka den experimentella datan resulterade i ett antal avgränsningar. Algoritmen som identifierar EPSP resulterade i avgränsning vad gällande vilken typ av information om EPSP som samlades in. Algoritmen samlade enbart in information om stigningens storlek i tids- och potentialled. Detta berodde på att fallbeteendet ofta var dolt i mätningarna. Dessutom exkluderades små EPSP från undersökningarna för att undvika att identifiera brus som EPSP, och således försäkra att informationen om EPSP enbart innefattar EPSP.

Andra avgränsningar som gjordes var att tröskelvärdet och intensitetsparametrarna behandlades som konstanta. Dessutom så valdes en brusmodell som är semideterministisk. Ankomsten av EPSP modellerades som en Poissonprocess vilket medför antagandet om att ankomsterna är oberoende av varandra.

4

Resultat

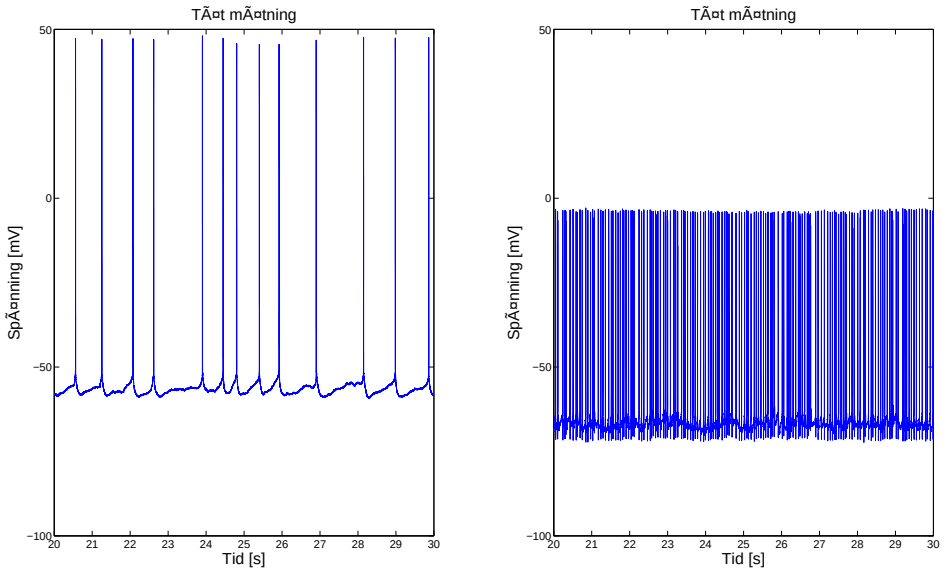
I föregående kapitel beskrevs egenskaper hos spikar, EPSP och brus samt hur dessa egenskaper undersöktes från mätdatan. Här presenteras hur modellen för neuronen konstruerades och hur de olika delarna modellerades. En jämförelse mellan modellen och faktiskt mätdata visas och statistiska undersökningar redogörs.

4.1 Statistiska undersökningar

I det här avsnittet presenteras värden på data, olika parametrar som tagits fram med fördelnings- och funktionsanpassning och resultat från statistiska undersökningar som beskrevs i kapitel 3.

De 18 olika mätningarna som undersökts varierar kraftigt i utseende. Grundnivån som beräknades för de 18 olika mätningarna från Sahlgrenska, hade ett medelvärde på $-64{,}6\,\text{mV}$ och standardavvikelsen var $6{,}5\,\text{mV}$. 11 av mätningarna hade spikar i sig och utav dessa hade tre stycken över 1000 räknade spikar i sig medan medelvärdet av antal spikar i de övriga åtta mätningarna var 61 stycken. Skillnaden i spiktäthet kan ses i figur 4.1. Medelvärdet för tröskelvärdet för mätningarna som hade spikar var $10{,}4\,\text{mV}$ ovan sina grundnivåer med standardavvikelsen $2{,}4\,\text{mV}$.

Vid undersökningen av hur spikarnas amplitud varierar, som presenterades i avsnitt 3.1.2, hittades ett samband med hur stort tidsintervallet var mellan två närmsta spikar. Detta kan ses i figur 4.2 där spikar som genereras mycket tätt tenderar att sjunka i amplitud proportionellt mot $\frac{1}{\Delta t}$. Vid längre tidsintervall tycks detta samband inte existera. Istället hittades ett samband via hur den föregående amplituden avvek från mätningens medelamplitud. I figur 4.2 representeras detta i form av färgskalan. De här två sambanden kunde uttryckas i form av en funktionsyta, där funktionen beter sig som en potensfunktion för små tidsavstånd och som ett plan för större avstånd. Planet uttrycker att spikar som har en höjd skild från medelvärdet, större eller mindre, tenderar att röra sig närmare detta värde. Funktionen som anpassades hade $R^2_{adj} \approx 0.91$ och går att betrakta mer noggrant i bilaga D.1. Datan är hämtad endast från de tre mätningar med flest spikar i sig, vilket medför att de längre tidsintervallen som presenteras är inte de längsta intervallen som har observerats. De övriga mätningarna hade för få spikar för att se något speciellt samband.

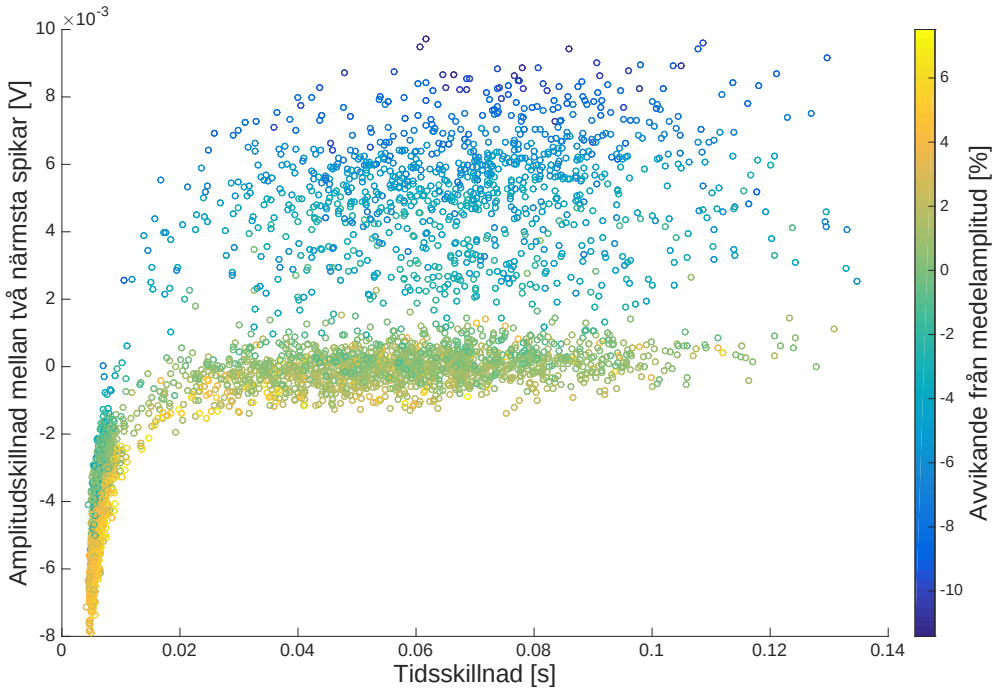


Figur 4.1: Figuren visar tio sekunder av två olika potentialmätningar med olika spiktätheter. I den glesa mätningen till vänster var medellängden för tidsintervallen mellan spikarna 782 ms, standardavvikelsen 207 ms. Den hade totalt 77 spikar i sig. I den tätare mätningen var medelintervallet 57,7 ms med standardavvikelsen 28,1 ms, en mätning med 1039 spikar i sig.

Ett liknande samband hittades för hur efterpotentialernas amplitud varierar. I figur 4.3 visas att korta tidsintervall medför en minskning hos den följande efterpotentialen, samt att avvikelsen från medelamplituden (färgskalan) har betydelse för längre tidsskillnader. Av samma skäl som för spikarna är datan i figur 4.3 endast tagen från de mätningar med flest spikar. Funktionen som anpassades hade $R^2_{adj} \approx 0{,}87$ och beskrivs mer ingående i bilaga D.2.

De mätningar som saknade spikar varierade både gällande tidsintervall mellan EPSP och även när det gäller amplituden hos dessa. I figur 4.4 visas urklipp från den mätning med flest EPSP utan spikar (till höger i figuren) och den mätning med minst antal EPSP. Det framgår även i figuren hur amplituden på EPSP varierar. Resultaten av fördelningsanpassningarna relaterade till EPSP presenteras nedan. Då många mätningar antingen var fulla med spikar eller var för brusiga för att identifieringsalgoritmen skulle fungera ordentligt så kunde data endast hämtas från fem mätningar.

Som kan ses i figur 4.5 anpassades stigtiderna enligt gammafördelningar. Hos fyra av mätningarna låg k-parametern mellan 6,4 och 8,6, medan den femtes (den med minst antal identifierade EPSP) hade ett mer extremt värde, 20,8. Även för θ-parametern skiljer sig denna mätning från resten (om än ej till samma grad) med ett värde på $3{,}8 \times 10^{-4}$ s. De andras låg mellan $7{,}2 \times 10^{-4}$ s och $10{,}7 \times 10^{-4}$ s.

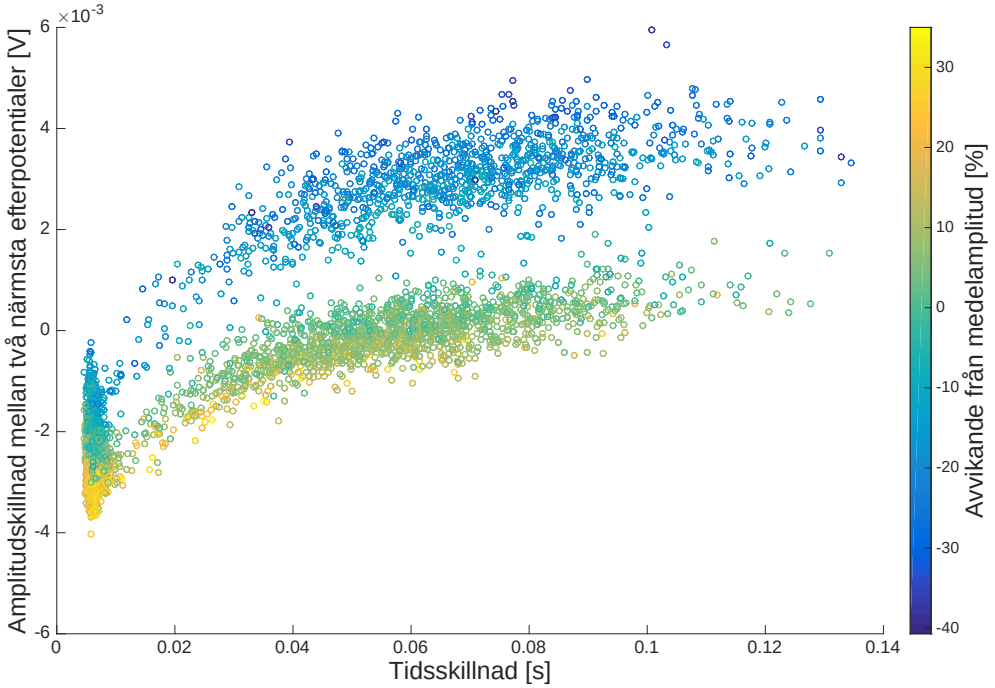


Figur 4.2: Amplitudskillnad mellan två närmaste spikar som funktion av tidsintervallet mellan dem samt den föregående spikens procentuella avvikelsen från medelamplituden (färgskalan). Datan är hämtad från de tre mätningarna som har flest spikar. Notera att större negativ avvikelse från medelvärdet tenderar en större positiv amplitudskillnad. En annan sak att uppmärksamma är att amplitudskillnaden visar sig bli mer negativ då tiden mellan spikar är mycket liten. Med denna data gjordes en funktionsanpassning som modellerar amplitudskillnaden för spikar.

Den mer extrema fördelningskurvan hade alltså en större förskjutning i tidsled än de andra; kurvan var centrerad kring 8 ms till skillnad från de andras ca 5 till 6 ms. Samtliga anpassningar klarade χ^2-testen med p-värden mellan 0,10 och 0,72.

Motsvarande p-värden för anpassningarna till det X som definierades i (3.3) låg mellan 0,11 och 0,75. Den mätning som var avvikande gällande stigtiderna var inte lika extrem i det här fallet (den hade dock lägst parametrar). För k-parametern låg värdena mellan 1,1 och 3,0 medan θ-parametrarna låg mellan $1{,}6 \times 10^{-4}\,\mathrm{V}$ och $6{,}0 \times 10^{-4}\,\mathrm{V}$. Dessa fördelningar var alltså inte alls lika förskjutna som stigtidernas. Denna information kan ses i figur 4.6.

Den regression som används i amplitudmodellen kan ses i figur 4.7. Den linjära korrelationen mellan amplitud och stigtid låg på ett rätt smalt intervall mellan 0,55 och 0,63. Första- och nolltetermskoefficienterna för regressionen låg mellan $0{,}79 \times 10^{-1}\,\mathrm{V\,s^{-1}}$ och $2{,}3 \times 10^{-1}\,\mathrm{V\,s^{-1}}$ respektive $0{,}18 \times 10^{-4}\,\mathrm{V}$ och $2{,}8 \times 10^{-4}\,\mathrm{V}$. Det fanns alltså viss variation i lutningen som definitivt kan spela roll för amplituden hos genererade EPSP.

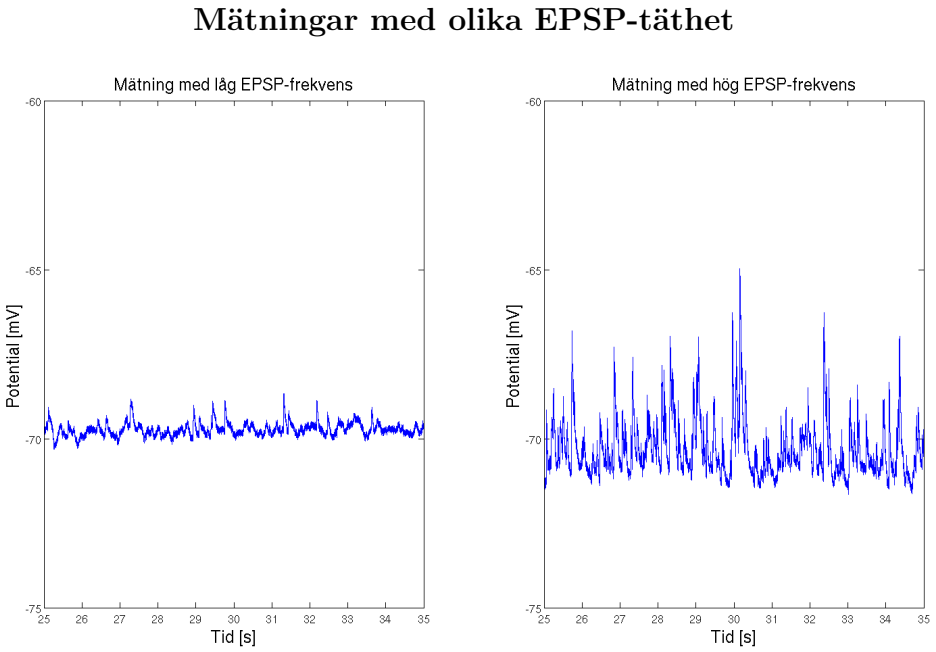


Figur 4.3: Amplitudskillnad mellan två närmaste efterpotentialer som funktion av tidsintervallet mellan dem samt den föregående efterpotentialens procentuella avvikelsen från medelamplituden (färgskalan). Datan är hämtad från de tre mätningarna som har flest spikar. I likhet med spikarna tenderar en större positiv amplitudskillnad gälla vid större negativ avvikelse från medelvärdet. Även då efterpotentialerna kommer mycket tätt visar det sig att amplitudskillnaden blir mer negativ. Med denna data gjordes en funktionsanpassning som modellerar amplitudskillnaden för efterpotentialer.

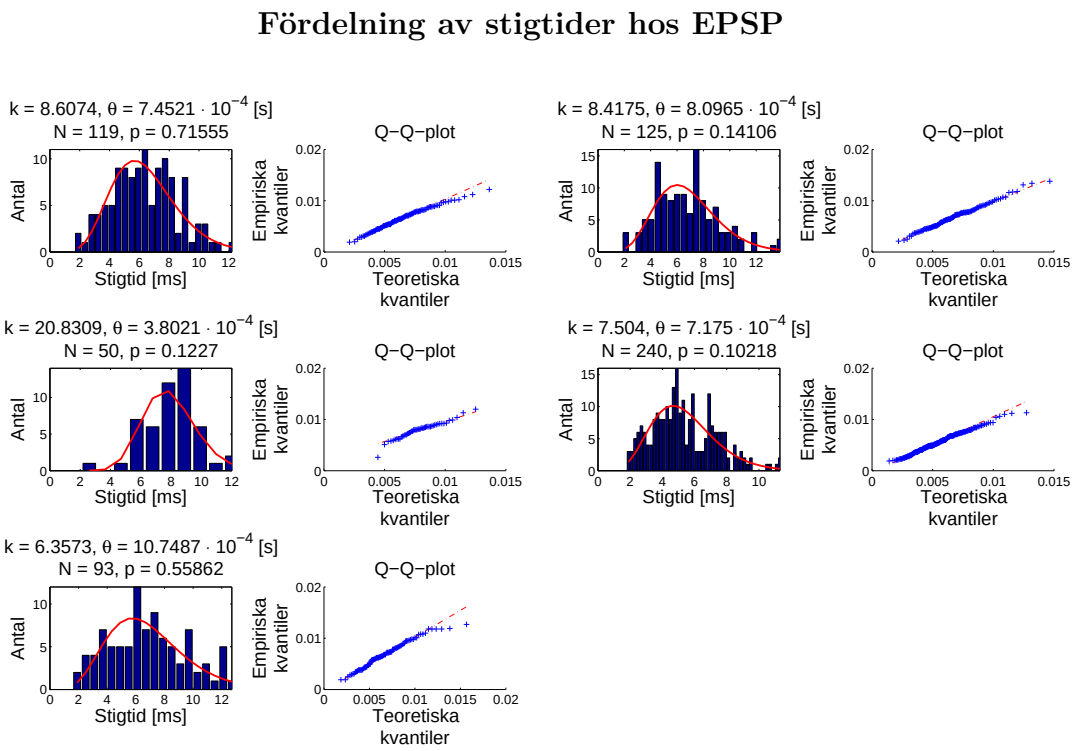
Konstanten c som behövdes för att göra X icke-negativ låg mellan $-0{,}17 \times 10^{-3}\,\mathrm{V}$ och $-1{,}1 \times 10^{-3}\,\mathrm{V}$. Ingen korrelation hittades dock mellan längd på tidsintervall och vare sig amplitud eller stigtid. Den uppmätta amplituden i sig hade medelvärden och standardavvikelser mellan 0,72 mV och 1,7 mV respektive 0,22 mV och 0,95 mV.

Som visas i figur 4.8 kunde tidsintervallen anpassas efter en exponentialfördelning. Av denna anledning kunde både χ^2- och Lillieforstestet användas. χ^2-testet gav p-värden mellan 0,25 och 0,78 medan Lillieforstestets värden låg mellan 0,14 och 0,50. Det sista värdet bör egentligen vara högre, men MATLAB:s Lillieforstest saknar tabellerade värden för högre p-värden utan att använda sig av simuleringsmetoder. Intensitetsparametrarna varierade naturligtvis beroende på hur tätt de identifierade EPSP var placerade, och låg mellan $1{,}00\,\mathrm{s}^{-1}$ och $13{,}0\,\mathrm{s}^{-1}$.

Rent allmänt så täckte fördelningsanpassningarna in den övergripande formen hos datamängderna, även om det fanns vissa avstickare. Detta syns tydligast i många av Q-Q-plottarna, där det kan ses att de flesta punkter, men inte alla, bildar en rätt linjär kurva.

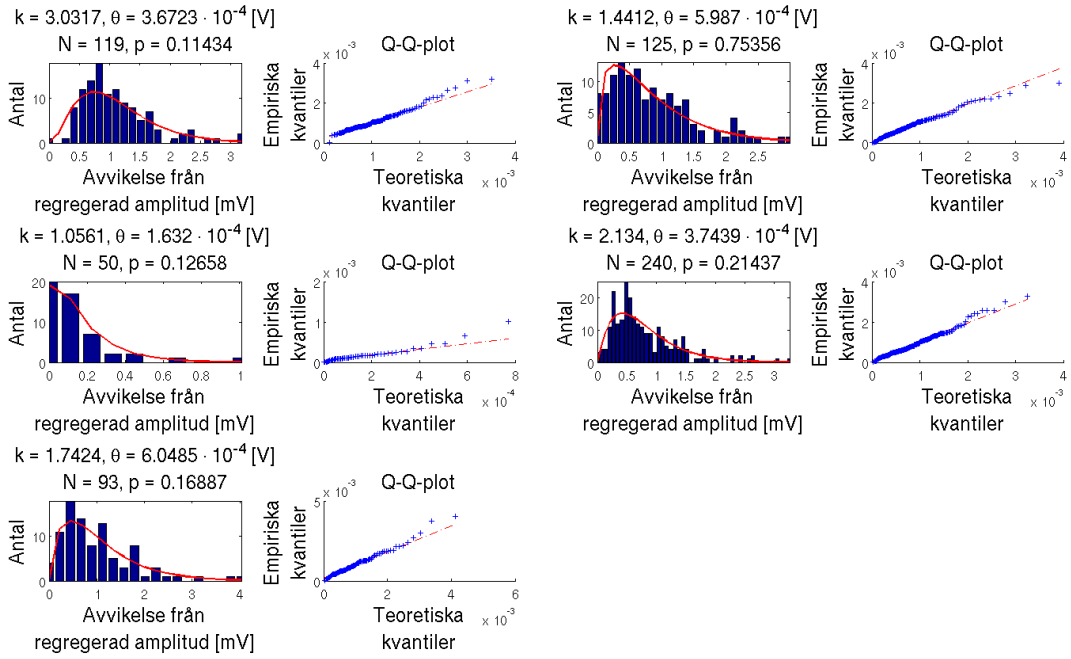


Figur 4.4: Figuren visar tio sekunder av två olika spänningsmätningar som saknar spikar. Den till vänster har minst antal räknade EPSP och den till höger flest EPSP utan att ha spikar. De olika intensitetsparametrarna tillhörande fördelningsanpassningarna var $13{,}0\,\mathrm{s}^{-1}$ respektive $0{,}997\,\mathrm{s}^{-1}$.

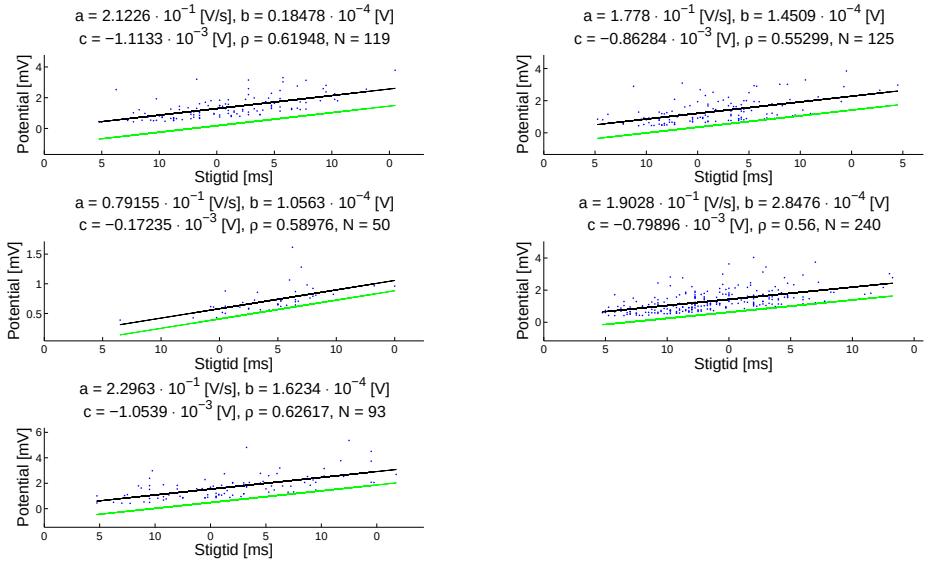


Figur 4.5: Figur som visar fördelningen av stigtider hos fem olika neuroner med N identifierade EPSP var. Stigtiderna har anpassats efter en gammafördelning med parametrar k och θ. Dessa uppskattades med Maximum Likelihoodmetoden. Samtliga anpassningar har klarat χ^2-testet med signifikansnivå 0,95. I både histogrammet och i χ^2-testet användes $\frac{N}{5}$ intervall, avrundat till närmaste heltal. Bredvid histogrammen ligger tillhörande Q-Q-plottar där jämförelsen gjorts mellan mätdatan och simulerade värden med hjälp av de uppskattade parametrarna.

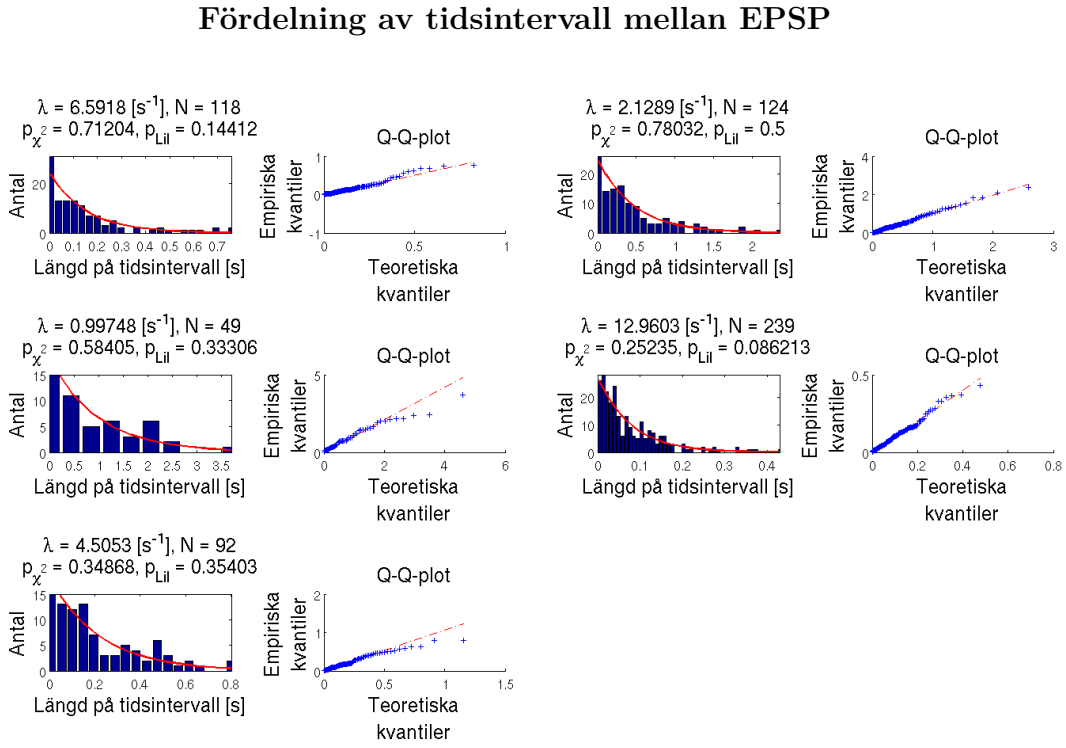
Fördelning av den stokastiska variabeln X som relateras till amplitud hos EPSP



Figur 4.6: Den fördelning som denna figur visar är för det X som definieras i (3.3), alltså skillnaden mellan uppmätt amplitud hos EPSP och den som fås genom en linjär regression av amplitud som funktion av stigtid (så när på en konstant). Datan har tagits från fem olika neuroner med N identifierade EPSP var. Värdena har anpassats efter en gammafördelning med parametrar k och θ. Samtliga anpassningar har klarat χ^2-testet med signifikansnivå 0,95. I både histogrammet och i χ^2-testet användes $\frac{N}{5}$ intervall, avrundat till närmaste heltal. Bredvid histogrammen ligger tillhörande Q-Q-plottar där jämförelsen gjorts mellan mätdatan och simulerade värden med hjälp av de uppskattade parametrarna.



Figur 4.7: Figuren visar amplitud mot stigtid hos EPSP från fem olika neuroner med N identifierade EPSP var. Den visar även en linjär regression $at_{stig}+b$ (den övre svarta linjen) samt denna translaterad med en konstant c (den undre gröna linjen). Denna regression är den som definierats i (3.3). Den linjära korrelationskoefficienten mellan amplitud och stigtid beskrivs av ρ.



Figur 4.8: Figur som visar fördelningen av längd på tidsintervall mellan EPSP hos fem olika neuroner med $N+1$ identifierade EPSP var. Intervallängderna har anpassats efter en exponentialfördelning med parameter λ. Denna parameter uppskattades med Maximum Likelihoodmetoden. Samtliga anpassningar har klarat både χ^2- och Lillieforstesten med signifikansnivå 0,95. Att ett av p-värdena är exakt 0,5000 beror på att MATLAB ej har p-värden över 0,5 tabellerade; för att få en bättre uppskattning krävs simuleringsmetoder. I både histogrammet och i testerna användes $\frac{N}{5}$ intervall, avrundat till närmaste heltal. Bredvid histogrammen ligger tillhörande Q-Q-plottar där jämförelsen gjorts mellan mätdatan och simulerade värden med hjälp av de uppskattade parametrarna.

4.2 Modellens konstruktion

Här presenteras modellens algoritm som baseras på de statistiska undersökningarna i föregående avsnitt. Först redogörs en helhetsbild och sedan behandlas modellens delar som spik, EPSP, och brus var för sig.

4.2.1 Översikt

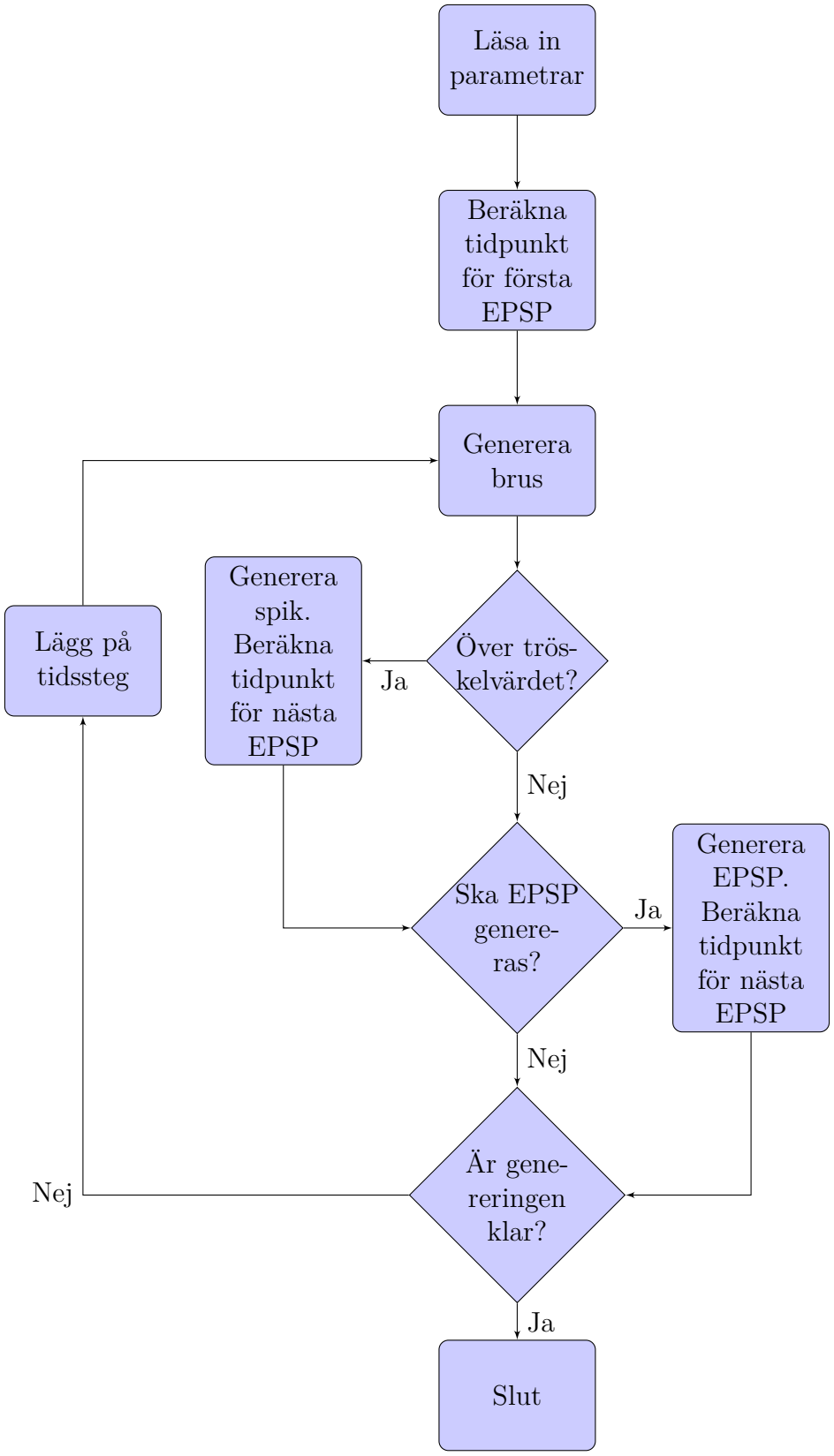
Den slutgiltiga modellen byggs upp genom separata modeller av spikar, EPSP, och brus. I en simulering läses först 14 parametrar in. Dessa kan ses i listan nedan.

- **Grundnivå:** Den potential som brus, EPSP och spikar utgår från.
- **Tröskelvärde:** Om potentialen överskrider detta värde genereras en spik.
- $\mathbf{k_{stig}}, \theta_{\mathbf{stig}}, \mathbf{k_X}, \theta_{\mathbf{X}}, \mathbf{a}, \mathbf{b} + \mathbf{c}$**:** Parametrarna k och θ är form- och skalparametrar för de gammafördelningar som styr stigtid och amplitud för EPSP. Parametrarna a och $b + c$ hör till den linjära regression som bestämmer amplituden hos en EPSP som funktion av dess stigtid.
- λ**:** Denna intensitetsparameter styr hur tätt EPSP kommer.
- $\mathbf{t_{stig_{spik}}}, \mathbf{A_{medel_{spik}}}, \mathbf{t_{fall_{spik}}}$**:** Dessa parametrar beskriver stig- och falltid och medelamplitud för spikar.
- $\mathbf{t_{fall_{efter}}}, \mathbf{A_{medel_{efter}}}$**:** Dessa parametrar beskriver falltid och amplitud för efterpotentialen.

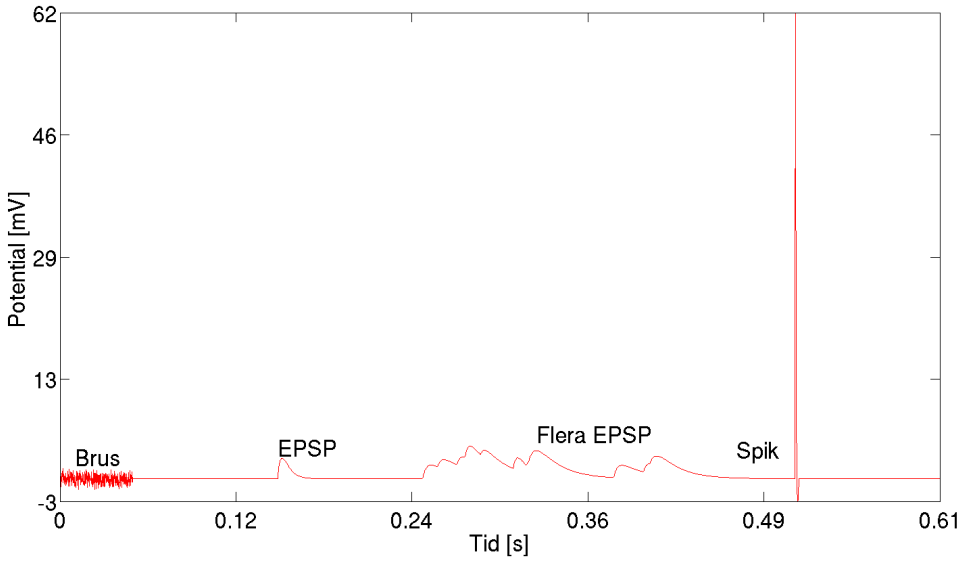
Ur programmeringssynpunkt behövs också hur lång tid simuleringen ska pågå och hur många punkter per sekund som används. I samtliga simuleringar har samplingsfrekvensen 10 kHz använts som detta värde. Alltså samma samplingsfrekvens som mätningarna är utförda med.

Potentialen sätts i början till grundnivån. EPSP av olika former fördelas över tidsintervallet som en Poissonprocess med intensitetsparametern λ, och positionerna sparas i en positionsvektor.

I figur 4.9 visas ett flödeschema över hur simuleringenprocessen ser ut. För varje tidssteg läggs brus på. Om den totala potentialen överstiger tröskelvärdet så genereras en spik med en följande efterpotential. En EPSP genereras vid varje tidpunkt i positionsvektorn. Om flera EPSP inträffar tätt kommer de att läggas på varandra. Det kan inte uppkomma några EPSP under spikgenereringen, utan först då efterpotentialen har nått sin minimala punkt. Utseendet hos spikar, EPSP, och brus bestäms av olika funktioner som har sina respektive parametrar som indata. Utseendet på brus, en enstaka EPSP, flera EPSP lagrade på varandra och en spik finns att ses i figur 4.10.



Figur 4.9: Flödesschema över modellens algoritm.



Figur 4.10: Figuren visar generering av brus, en enstaka EPSP, ett tåg av flera EPSP och en spik. Notera att alla fyra är uppritade oberoende av varandra.

Eftersom brusfunktionen till viss del är deterministisk kan inte bruset fortsätta som vanligt efter en spik utan risk för stora diskontinuiteter. Om brus skilt från noll skulle lagras på potentialen skulle den alltså göra ett stort hopp. Efter en spik startar istället brusfunktionen på nytt i en punkt där den är approximativt noll.

4.2.2 Modellering av spik och efterpotential

I [8] beskriver Hodgkin-Huxleymodellen hur en spik (och resten av potentialen) kan modelleras efter differentialekvationer som beskriver jonflöden, speciellt natrium- och kaliumflödet i neuronen. En sådan modell ger ett väldigt realistiskt utseende på spiken och tillåter studier av störningar hos jonbalanser och flöden. Nackdelen med Hodgkin-Huxleymodellen är att den beskrivs av fyra differentialekvationer som behöver lösas parallellt, och saknar därför en analytisk lösning. Därför är det svårt att på ett enkelt sätt styra exakt hur amplitud, stigtid och falltid varierar både för spiken och efterpotentialen. I detta arbete fokuseras inte på jonflöden, utan snarare på statistik hos olika beteenden. Därför används en betydligt enklare modell.

Utseendet hos spikarna modelleras med en så kallad lorentzianfunktion på formen

$$A \cdot \frac{\gamma^2}{(t - t_0)^2 + \gamma^2}, \tag{4.1}$$

där A är maximala amplituden som inträffar vid tiden t_0 och γ är en formparameter. I vår modellering sätts värdet på formfaktorn γ till en tredjedel av stigtiden före tiden t_0 och en tredjedel av falltiden efter tiden t_0. Detta beror på definitionen av γ och dess relation till dessa tider. Att γ ändrar värde medför att karaktäristiken under stig-

och falltid efterliknas. t_0 sätts helt enkelt till stigtiden. Lorentzianfunktionen valdes främst för att den har enkla parametrar för att bestämma höjd och bredd.

Spikens amplitud är definierat relativt tröskelvärdet. Amplituden beräknas av en funktion som tar fram amplitudändringen jämfört med föregående spik. Funktionen har tidsskillnaden mellan spikarna (Δt) samt den föregående spikens procentuella avståndet till en medelamplitud ($\frac{A_{före}-A_{medel}}{A_{medel}}$), som indata och ges av

$$\Delta A_{spik} = \frac{a_1}{\Delta t} + b_1 \cdot \frac{A_{före} - A_{medel}}{A_{medel}} + c_1 \quad [\mathrm{V}], \tag{4.2}$$

med konstanterna a_1 $=-2{,}769 \times 10^{-5}\,\mathrm{V\,s}$, b_1 $=-0{,}061\,13\,\mathrm{V}$, och c_1 $=0{,}001\,76\,\mathrm{V}$. Konstanterna kommer ifrån MATLAB:s *cftool*. En konsekvens med denna funktion är att även en stor amplitud kommer att följas med en mindre, vilket är ett beteende som inte har kunnat observerats.

Genereringen av efterpotentialen börjar när spiken har sjunkit ner till tröskelvärdet. Formen är inspirerad från den radiella delen av vågfunktionen hos en väteatom enligt

$$V = -Cte^{-\frac{t}{T}} \quad [\mathrm{V}], \tag{4.3}$$

där V är potentialen, t tiden, och C,T är positiva konstanter som formar funktionen. Konstanten T är den tid då efterpotentialens amplitud är som maximal, $T = t_{max}$. Konstanten C bestäms genom hur stor amplituden är, $C = \frac{V_{max} \cdot e^1}{T}$.

Liknande spikens amplitud är efterpotentialens amplitud definierat relativt tröskelvärdet, men positivt. Denna amplitud fås också fram genom en funktion som ger ut amplitudskillnaden jämfört med den förra efterpotentialen. Inparametrarna är som innan tidsintervallet och den föregående amplitudens procentuella avstånd till en medelamplitud för efterpotentialen enligt

$$\Delta A_{efterpotential} = \frac{a_2}{\Delta t} + b_2 \cdot \frac{A_{före} - A_{medel}}{A_{medel}} + c_2 \quad [\mathrm{V}], \tag{4.4}$$

med konstanterna a_2 $=-1{,}776 \times 10^{-5}\,\mathrm{V\,s}$, b_2 $=-0{,}008\,532\,\mathrm{V}$, och c_2 $=0{,}001\,114\,\mathrm{V}$. Konstanterna kommer ifrån MATLAB:s *cftool*.

4.2.3 Modellering av EPSP

Antalet EPSP inom ett visst tidsintervall modelleras som en homogen Poissonprocess genom att placera ut dem med exponentialfördelade tidsavstånd.

Varje enskild EPSP genereras av uttrycket

$$\frac{w}{\tau - \tau_r}\left(e^{-\frac{t}{\tau}} - e^{-\frac{t}{\tau_r}}\right), \tag{4.5}$$

där w är en skalfaktor och τ_r och τ är tidskonstanter som beräknas beräknas utifrån stigtid och amplitud. τ_r är membranets passiva tidskonstant och τ är den synaptiska tidskonstanten, båda vilka introducerades i avsnitt 2.2. Uttrycket kommer från

Burkitts artikel om IFM [17]. Det används också av TempUnit, ett datorarkitektur som använder sig av neurala nätverk för signalhantering [32].

Genom att derivera detta uttryck kan fås att stigtiden t_r ges av

$$t_r = \frac{\tau_r \tau}{\tau_r - \tau} \ln \frac{\tau_r}{\tau}. \tag{4.6}$$

Eftersom stigtiden här är definierad som tiden från att EPSP:n startar till det att den nått sitt maxvärde A så gäller även

$$A = \frac{w}{\tau - \tau_r} \left(e^{-\frac{t_r}{\tau}} - e^{-\frac{t_r}{\tau_r}} \right). \tag{4.7}$$

Insättning av (4.6) i (4.7) ger då

$$w = A \frac{\tau - \tau_r}{\left(\frac{\tau_r}{\tau}\right)^{\frac{\tau_r}{\tau - \tau_r}} - \left(\frac{\tau_r}{\tau}\right)^{\frac{\tau}{\tau - \tau_r}}}. \tag{4.8}$$

Dessa tre ekvationer bildar ett ekvationssystem med de okända variablerna w, τ och τ_r. Detta system saknar analytisk lösning, och behöver därför lösas numeriskt för varje par av genererade värden för t_r och A.

4.2.4 Modellering av brus

Det visade sig att det fanns många gemensamma frekvenstoppar för de olika mätningarna när de studerades i frekvensplanet. De frekvenser som modellerar bruset samt dess amplitud går att se i tabell B.1 i bilaga B. Efter att frekvenserna har satts ihop i en summa av sinustermer lades en slumpmässig störning på för varje mätpunkt enligt

$$d = n_1(2n_2 - 1) \cdot 20 \times 10^{-6} \quad [\text{V}] \tag{4.9}$$

där d är störningen och n_1, n_2 är slumpmässiga tal mellan 0 och 1 som genereras för varje mätpunkt.

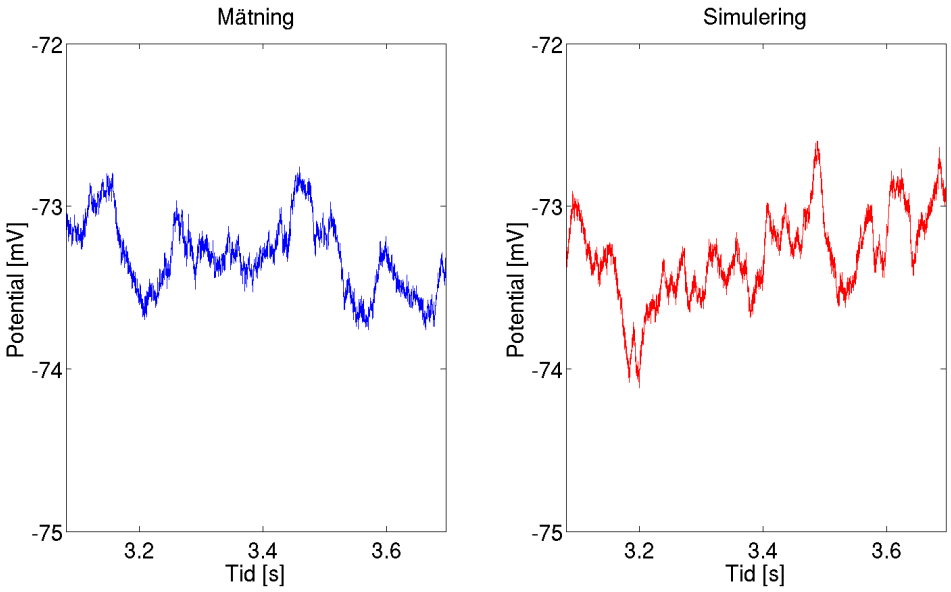
4.3 Jämförelse mellan modell och data

Att göra några kvantitativa jämförelser mellan modellen och datan lyckades inte. I stället jämfördes olika egenskaper och beteenden kvalitativt. Detta presenteras nedan.

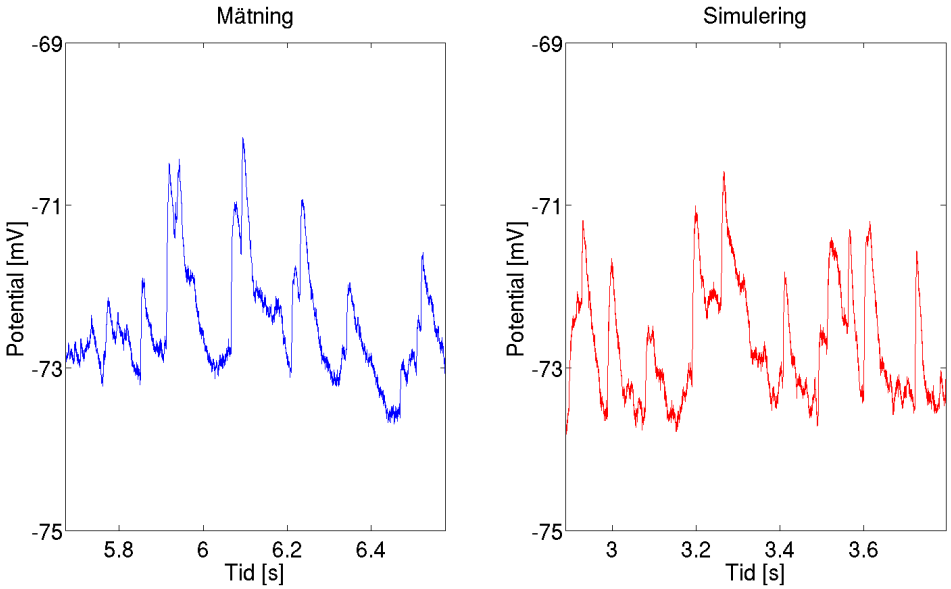
En jämförelse mellan uppmätt och genererat brus kan ses i figur 4.11. Observera att det inte finns någon garanti för att bruset från datan inte innehåller någon EPSP som varit för liten för algoritmen att hitta.

Ett segment av datan som innehåller både EPSP och brus kan ses bredvid en motsvarande generering i figur 4.12. De parametrar som användes för att generera EPSP kommer från den mätning de jämförs med, men har modifierats för att bli mer lik datan (enligt ögonmått).

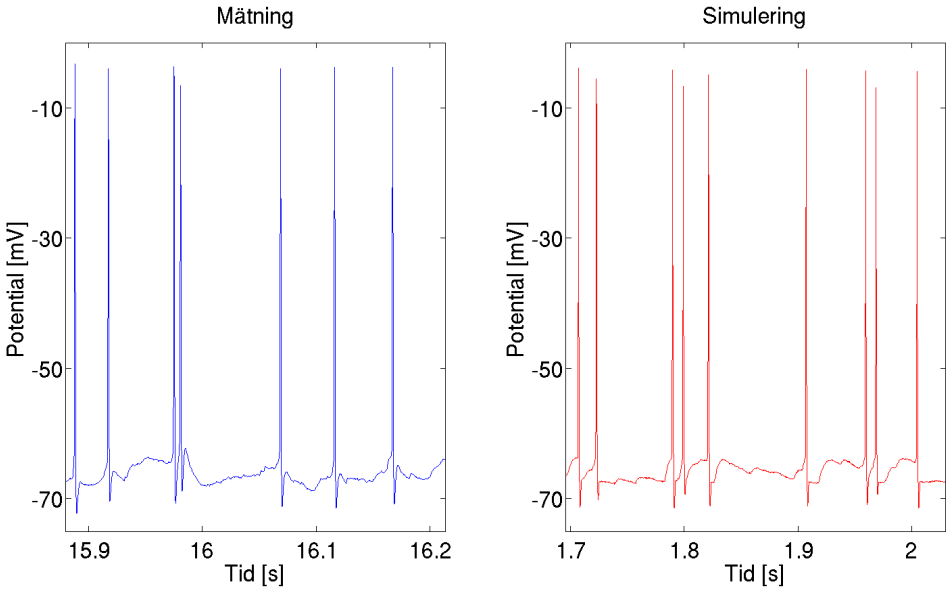
Figur 4.13 visar en jämförelse mellan ett antal spikar som genererats av modellen och spikar från datan. I figuren kan man bland annat se hur både spikens och efterpotentialens amplitud beror på avståndet till föregående spik (eller efterpotential) samt dennas amplitud.



Figur 4.11: Ett segment av brus från datan kan ses i den vänstra grafen medan den högra innehåller brus som genererats. Det finns dock ingen garanti för att bruset från datan verkligen endast består av brus och inte innehåller någon EPSP som varit för liten för att detekteras.



Figur 4.12: I den vänstra grafen kan en sekvens av EPSP ses, medan den högra innehåller en motsvarande simulerining. Parametrarna som användes vid simuleringen är baserade på de som tagits från datan i den vänstra grafen, men modifierade för att ge ett mer likt utseende.



Figur 4.13: Denna figur visar en jämförelse mellan spikar funna i datan (vänster) och spikar genererade av modellen (höger). En egenskap hos båda är hur amplituden beror på den hos och avståndet till förekommande spik. Det samma gäller för storleken på efterpotentialen. Det finns även vissa skillnader, exempelvis hur modellen saknar den ökning över grundnivån som sker efter efterpotentialen.

5

Diskussion

Den matematiska modellen av potentialen som har presenterats framhäver beteenden som är karakteristiska för en neuron. Dock är den inte perfekt. Nedan diskuteras resultatet av projektet, undersökningar av data, på vilka sätt som modellen fungerar och på vilka sätt den hade kunnat förbättras.

5.1 Val av modeller

Valet av modell baserades dels på undersökning av mätdata och dels på litteraturstudien som gjordes vid projektets start. Med tanke på karaktären hos den datan vi undersökte så var en IFM-liknande modell lätt att implementera utifrån undersökta parametrar. Den biofysikaliska delen av vår modell är att neuronen har ett tröskelvärde och att om potentialen, genom mottagning av EPSP, når över tröskelvärdet genereras en spik. Detta är något som också används vid IFM. Genom att utgå från den aspekten påbörjades undersökningar av EPSP i syfte att hitta en fördelning för EPSP:s ankomst under antagandet att denna skulle kunna beskrivas stokastiskt, också i enlighet med IFM.

Så som beskrivet i avsnitt 4.2.2, valdes inte en biofysikalisk modell för att generera spikutseende. Hodgkin-Huxleymodellen som ska utgöra en god beskrivning för spikdynamiken kräver information om jonflödet vilket vi inte undersökte. Att anpassa biofysikaliskt data efter de parametrar som faktiskt undersöktes hade varit komplicerat om än intressant. Den skarvande lorentzianen var däremot väldigt anpassningsbar till de uppmätta parametrarna och gav ett tillfredsställande utseende, även om den inte nödvändigtvis har någon fysikalisk motivering.

Vad gäller amplituden hos spikarna och efterpotentialerna som behandlades i avsnitt 4.2.2, så bestäms de av en deterministisk funktion och är inte så slumpmässiga som de borde vara. Enligt mätdatan fanns det en spridning från den funktionsyta som räknats fram. Detta hade kunnat modifieras genom att lägga på en sannolikhetsfördelning på funktionen. Däremot var det svårt att ta fram en sådan fördelning som överensstämde väl med datan, just på grund av att mätdatan i sig inte uppvisade en tydlig fördelning. Mätdatan visade tendenser till att ha fördelningar som varierade beroende på var på funktionsytan man tittar. Det hade varit möjligt att utveckla en modell som har en varierande fördelning, men det hade krävt mer data. En deterministisk modell räckte för att modellen skulle få ett till synes realistiskt utseende.

Spikarna har trots allt en slumpmässig placering på grund av deras EPSP-beroende, och ger därför den deterministiska modellen en slumpmässig prägel.

Något som var lite mer fysikaliskt motiverat än lorentzianmodellen var uttrycket för generering av en enstaka EPSP, (4.5), i och med att det är baserat på IFM. Den har däremot ett par praktiska nackdelar. Den första är ett problem som den delar med Hodgkin-Huxleymodellen, nämligen att det inte går att hitta ett analytiskt uttryck för de parametrar som styr utseendet som funktion av amplitud och stigtid. Detta arbetas runt genom att lösa ett icke linjärt ekvationssystem numeriskt för varje enskild EPSP, vilket definitivt är görbart men förlänger tiden det tar för simuleringen att köras.

Något som hade kunnat göras för att åtgärda detta är att använda en annan EPSP-modell, exempelvis en med samma typ av utseende som efterpotentialen till en spik enligt (4.3). För denna kan ett analytiskt samband hittas på samma sätt som för efterpotentialen, och den har ett utseende som påminner om en EPSP. Den saknar dock den fysikaliska motiveringen som den modell som används har.

En annan möjlig angreppsvinkel skulle vara att försöka göra en fördelningsanpassning av de parametrar som fås om samtliga par av stigtid och amplitud körs genom ekvationssystemslösaren. Problemet med denna metod är att även om det skulle lyckas så är det betydligt mindre intuitivt vad parametrarna τ, τ_r och w från (4.5) faktiskt innebär än vad stigtid och amplitud gör.

Det andra problemet med EPSP-genereringen är att det inte går att kontrollera både stig- och falltiden samtidigt, ett problem som gäller även för den efterpotentialbaserade modellen. I denna studie lyckades inget lämpligt sätt att studera falltid tas fram, så det finns i nuläget inget sätt att undersöka om falltiden för modellgenererade EPSP stämmer överens med empiriska värden på ett kvantitativt sett. Behovet att kunna kontrollera både stig- och falltid separat kan komma att behövas om modellen vidareutvecklas.

Att just gammafördelningen valdes till stigtid och X som definierad i (3.3) har ingen annan motivering än att den såg ut att kunna fungera, figur 4.5 och 4.6. Även normal- och exponentialfördelning testades för vissa datamängder, men de var ofta för asymmetriska för en normalfördelning och hade en maximipunkt för långt från 0 för att kunna vara en exponentialfördelning. Om någon annan fördelning skulle användas så är det förmodligen normalfördelningen för stigtiderna, med tanke på deras höga k-parametrar.

Utseendet på modelleringen av amplitudens beroende av stigtiden kom från att korrelationskoefficienten var för hög för att ignorera, men för låg för att implicera ett deterministiskt förhållande, se figur 4.7. Den modell som användes var därför en enkel ansats som kombinerade linjär korrelation med en stokastisk variabel. Denna variabel kunde visserligen anpassas efter en gammafördelning, men det finns ingen garanti för att ansatsen var den bästa som kunde göras. Som exempel hade även den stokastiska variabeln kunnat vara tidsberoende.

Att bruset har modellerats som en Fourierapproximation med vitt brus, som beskrivs i avsnitt 3.3, har både för- och nackdelar. En fördel är att det är väldigt lätt att

realisera. Dessutom kan det ses i figur 4.11 att det blir väldigt likt bruset i datan, i alla fall kvalitativt sett. En nackdel är att modellen blir mer deterministisk än den egentligen bör vara.

En annan nackdel är att brusfunktionen är anpassad specifikt utefter de mätningar som användes. Detta gäller naturligtvis all data som tagits fram, men det är lättare att ändra på exempelvis en parameter till en sannolikhetsfördelning för att kompensera för andra typer av neuroner än att ändra på ett flertal frekvenskomponenter.

En annan modell som i stället hade kunnat användas för bruset är någon form av slumpvandring (en typ av stokastisk process, exempelvis Brownsk rörelse). Detta testades men skulle kräva mer undersökning för att säkerställa att det var implementerbart och anpassat efter mätdatans brus.

Att använda en Poissonprocess för att placera ut EPSP motiveras till viss del i avsnitt 5.4, men själva ansatsen kan motiveras av att varje enskild neuron kan ha ett så stort antal synapser att de kopplade neuronerna kan modelleras som oberoende. Det är dock fullt möjligt att en inhomogen Poissonprocess, alltså en Poissonprocess där $\lambda = \lambda(t)$ är tidsberoende, hade varit ett bättre val.

Över långa tidsintervall är detta inte helt orimligt. Olika delar av hjärnan är trots allt olika aktiva över olika tidsintervall. Så länge λ inte varierar för kraftigt är dock en homogen process en bra approximation över ett tillräckligt kort tidsintervall. Dessutom är det betydligt mer komplicerat att undersöka om en datamängd kommer från en inhomogen Poissonprocess än från en homogen eftersom utseendet på de relevanta fördelningarna är beroende på vilket $\lambda(t)$ som ansätts.

5.2 Framtagning av data

I avsnitt 4.1 presenterades resultatet från undersökningar av mätningarna från Sahlgrenska universitetssjukhuset. Potentialernas grundnivåer bestämdes till viss del med ögonmått men de stämmer väl överens med litteraturen. Medelvärdet av potentialskillnaden mellan tröskelvärdena och grundnivåerna var i överensstämmelse med [11]. De beräknade tröskelvärdena kan dock förväntas vara underskattningar av de verkliga värdena då vi enbart använder oss av den högsta potentialen som inte utlöser en spik, och det är möjligt att det verkliga tröskelvärdet ligger en bit över det beräknade.

Att ta fram information och egenskaper hos spikar gick problemfritt. Tack vare att lorentzianfunktionen är så enkel att arbeta med när det gäller att forma utseendet hos spikarna behövdes endast stigtid, falltid och amplitud undersökas. Detsamma gäller även för efterpotentialerna. Vid studierna av hur amplituderna varierades användes endast de tre mätningarna som hade flest spikar i sig. Detta kan verka som något dåligt, men de mätningar med få spikar var så spridda att något tydligt samband kunde inte urskiljas. Som beskrivet i avsnitt 4.1 gick de tre mätningarna som användes att anpassa väl till en funktionsyta, vilket de höga R^2_{adj} indikerar. Samma

funktion går att använda till spikar som har såpass långa tidsavstånd att de inte påverkar varandra, eftersom funktionen får spikarna att röra sig kring medelamplituden och inte bort därifrån. På så sätt får man i någon mening en viss fördelning kring medelvärdet.

Med EPSP-algoritmen (figur 3.3) kunde en intensitetsparameter för tidsintervallsfördelningen skattas i de mätningar där algoritmen var brukbar. I de mätningar som hade hög spikfrekvens gick dock inte detta. En idé fanns om att variera intensitetsparametern tillhörande EPSP-genereringen och jämföra fördelningen för tidsintervallen mellan de generade spikarna och motsvarande fördelning för spikarna i mätningen. Emellertid gav inte fördelningsappassning till mätningarnas spikintervall något användbart resultat. Vidare försök gjordes att dela upp tidsintervallen mellan spikarna i mätningarna efter lokal spikfrekvens, i stil med undersökningen av Maimon och Assad [20], men undersökningen avbröts då antalet tidsintervall mellan spikar, då indelade enligt frekvensintervall, var för få för att dra någon egentlig slutsats.

Samma algoritm identifierade amplituder hos EPSP som till viss del ligger i samma storleksordning som i litteraturen. Den största amplituden är i samma storleksordning som i [14] medan det minsta värdet algoritmen hittar är ungefär tio gånger så stort som motsvarande minimivärde. Det var ett medvetet val att riskera att välja bort små EPSP för att på så sätt minska risken att få med brustoppar. Givet att algoritmen missar små EPSP skulle alltså de faktiska tidsintervallen i genomsnitt vara mindre än de uppmätta. Därav skulle de beräknade intensitetsparametrarna vara underskattningar av de verkliga.

Det är inte bara intensitetsparametern som blir felaktigt uppskattad. Eftersom små EPSP missas av algoritmen och amplitud verkar korrelera med stigtid så är det rimligt att anta att även vissa EPSP med korta stigtider har ignorerats. Med andra ord är det mycket möjligt att k-parametern har överskattats. Om denna parameter sänks i en simulering kan det dock vara vettigt att öka θ-parametern också, så att stigtiderna inte blir alltför korta.

Liknande slutsatser kan dras om parametrarna tillhörande X. Slutsatser kring regressionsparametrarna a och $b + c$ är dock inte lika lätta att dra. Exempelvis kan de inte varieras för mycket oberoende av varandra utan att ge en stor förändring av X.

Värt att nämna är att parametervärdena från den mätning med endast 50 identifierade EPSP inte bör tas lika seriöst som de från andra mätningar. Denna mätning har trots allt betydligt färre EPSP än någon av de andra, och den uppvisar utan tvekan de mest extrema parametervärdena.

EPSP:er var rent allmänt svåra att identifiera i mätningarna. Variationen i amplituden från nästan brusstorlek till markanta ökningar tillsammans med det faktum att de superponeras på varandra gjorde att framtagandet av en algoritm som skulle kunna identifiera EPSP svårt. Ett par olika metoder testades innan vi blev nöjda.

Utslätning av datat i syfte att ta bort brus gjorde att mindre EPSP kunde ha försvunnit. Dessutom kvarstod ett stort antal små toppar som förmodligen kom

från brus. Manuell undersökning testades på ett par mätningar men resulterade i under 100 identifierade tydliga EPSP. Dessutom var analysen av falltiden fortfarande osäker då det med ögonmått inte fanns någon tydlig sluttid för EPSP då risken alltid fanns att det kommit en till EPSP efter föregående som påverkade falltiden.

5.3 Modellens algoritm

Modellen har lyckats kombinera olika former och beteenden, och ritar generellt upp ett utseende som hade kunnat komma från en riktig mätning. Många stora buggar i simuleringskoden har lösts under arbetets gång. Det mest allvarliga var att de EPSP som skulle ha uppkommit under spikgenereringen inte försvann utan väntade till genereringen var klar. Detta resulterade att alla de EPSP skapades i princip samtidigt, och utlöste nästa spik. En annan bugg som också löstes var att den EPSP som kom över tröskelvärdet bidrog till potentialen efter spikgenereringen, vilket är ett beteende som inte har identifierats.

På grund av att varje EPSP genereras genom att lösa ett icke-linjärt ekvationssystem kommer långa simuleringar att behöva mycket datorkraft. Trots att algoritmen till viss del har optimerats tar det lång tid att beräkna potentialen. En simulering under 60 s och med $\lambda = 250\,\mathrm{s}^{-1}$ tar ungefär 3,5 minuter att köras. Då kommer approximativt 15000 EPSP att genereras. Detta kan jämföras med att antalet spikar i mätningarna varierar från ungefär 100 till 1700. Självklart varierar körningstiden också mellan olika hårdvara, men det går ändå att få en uppfattning om hur mycket arbetskraft som krävs. Om simuleringen gjorts i ett effektivare programspråk, till exempel C, så hade körningstiderna säkert förkortats betydligt. Fördelen med MATLAB är att datahanteringen är mycket enkel, och det är något som vi har värdesatt mycket. Körningstiden för att simulera en neuron är ändå acceptabel.

Om man skulle vilja konstruera ett nätverk av neuroner så skulle metodiken som presenterats i den här rapporten inte vara särskilt lämplig, då man snabbt skulle få mycket långa körningstider. Att bygga ett nätverk av neuroner och simulera dessa i realtid, för att till exempel skapa något form av intelligent system är inte att tänka på. I så fall skulle man behöva använda en betydligt enklare modell som enbart fokuserar på informationsöverföringen.

Simuleringen stegar fram i tiden och räknar ut potentialen för varje tidspunkt. Detta innebär att kurvan som skapas ska vara någorlunda kontinuerlig och inga större diskontinuiteter får finnas i potentialen. Detta fungerar mycket bra i modellen i och med att bruset har anpassats till att inte avvika i området där två olika funktioner sammanfaller. Kurvan är alltså analytiskt sett helt kontinuerlig om det vita bruset som hör till brusgeneratorn bortses från, men i praktiken kommer det naturligtvis finnas mindre diskontinuiteter på grund av datorers diskreta natur.

Trots att modellen inte har några större diskontinuiteter är övergången mellan att spiken slutar och efterpotentialen börjar något hackig. Detta beror på att potentialens derivata inte är kontinuerlig i detta område då det är två olika funktioner som

sätts samman. Detta är inte ett jättestort problem då vissa spikar visar ett liknande betende. Dock gäller det inte alla spikar.

Nackdelen angående determinismen hos brusmodellen kompenseras för till viss del genom att låta simuleringen börja i en slumpmässigt vald punkt i brusets period, och sedan stega sig fram. På detta sätt har inte bruset samma utseende för varje simulering. En liknande implementering vid spikgenereringen ser också till att spikarna blir olika.

5.4 Analys av modell

Modelleringen av ankomsten av EPSP som en Poissonprocess styrks av att det gick att anpassa tidsintervallen mellan identifierade EPSP med en exponentialfördelning, se figur 4.8. Under antagandet att algoritmen missar mindre EPSP stämmer inte tidsintervallsfördelngens parametrar med de verkliga, men den verkliga fördelningen bör ändå vara en exponentialfördelning givet att amplituden är slumpmässigt fördelad bland förekommande EPSP, det vill säga att amplituden mellan olika EPSP och tidsintervallen mellan dem är okorrelerade.

Det är dock viktigt att understryka att även om en Poissonprocess implicerar exponentialfördelade avstånd så gäller ej motsatsen. Att intervallen verkar vara fördelade på detta sätt styrker visserligen hypotesen att en Poissonprocess hade kunnat generera dem, men det är inte på något sätt ett bevis.

Något annat som hade kunnat undersökas för att styrka Poissonhypotesen är om antalet EPSP innanför ett visst tidsintervall t är Poissonfördelat med parameter λt. Detta var tyvärr inte görbart för majoriteten av mätningar, om ens för någon, med den data som fanns. Eftersom intensitetsparametern λ antas vara konstant krävs att samma mätning används för varje undersökning. Dessutom skulle intervallängden behöva vara så pass lång att tillräckligt många EPSP har möjligheten att hamna innanför intervallet. I och med detta blir det svårt att få plats med tillräckligt många tidsintervall på en mätning för att kunna säga något statistiskt signifikant om fördelningen.

Även om exponentialfördelade tidsintervall verkar rimligt så finns det dock ingen garanti för att övriga val av fördelningar är rimliga, speciellt när det rätt så garanterat finns EPSP som missats. De klarade visserligen de statistiska testerna, men p-värdena varierade ändå en del i storlek. Q-Q-plottarna (figur 4.8) uppvisade dock rätt så linjära beteenden, vilket stärker valet av fördelning. En intressant observation gällande dessa är att det framför allt är de senare kvantilerna som skiljer sig från mängden, vilket kan tolkas som att de typer av värden som modellen har svårast att återskapa är de största.

Ett typ av beteende som modellen saknar helt är förekomsten av IPSP. Detta beror på att inga sådana kunde hittas i datan, antingen för att de var för små för att kunna skiljas från bruset eller för att det helt enkelt inte fanns några. Det skulle dock vara väldigt enkelt att implementera IPSP-generering om ytterligare data användes

för att ta fram deras statistik, eftersom en IPSP i princip bara är en inverterad EPSP.

Att spikarnas amplitud minskar med korta tidsavstånd till föregående spik (figur 4.2) kan möjligtvis förklaras biologiskt genom att spikar som uppkommer tätt ihop inte hinner ladda upp sig fullständigt på den korta tiden.

Definitionen av spikamplitud relativt tröskelvärdet kan vara riskabel om man vill variera någon av dessa oberoende av varandra. Eftersom ökningen av potentialen vid en spik betyder att natrium passerar jonkanalen för att motverka koncentrationsgradienten över cellmembranet har aktionspotentialen en maximal amplitud då jämvikt för natrium har etablerats. Vid variation av amplitud eller tröskelvärde krävs uppmärksamhet så att spikamplituden inte blir ofysikaliskt hög.

I de spiktäta mätningarna kunde ett visst mönster i avfyrningen av spikar urskiljas. Spikarna kom tätt om tre följt av ett något längre tidsintervall och sedan upprepades mönstret. Denna karaktäristik går inte att efterlikna med modellen som den är nu. Antingen skulle man kunna införa en varierande intensitetsparameter eller kanske ännu hellre försöka modellera en mindre ökning av potentialen efter att den återgått till grundnivån. Detta för att ge utrymme för den effekt som efterliggande natriumströmmar kan ge och som kan orsaka ett tätt spiktåg. Vidare undersökningar av parametrarna hos den efterliggande potentialen skulle då behöva utföras.

Det genererade bruset är kvalitativt sett likt det uppmätta, men det finns en konsekvens av hur det togs fram som är viktig att ta upp. Eftersom bruset är skapat utifrån uppmätta frekvenskomponenter är det sannolikt att frekvenser från exempelvis mätutrustning inkluderats. Brusmodellen stämmer alltså väl överens med den experimentella data vi har, men är inte nödvändigtvis lika lik hur datan hade sett ut efter en ideal mätning. Det kan därför vara en bra idé att exkludera de frekvenser som kan antas ha icke-biologiska ursprung om modellen används till något annat än jämförelse med datan den är konstruerad från.

5.5 Slutsats

Modellen och algoritmen som har presenterats lyckades återskapa de grundläggande dragen och beteenden hos en neuron. Detta beror bland annat på att de mindre ingående delarna i modellen (spik, EPSP, brus) i sig har liknande egenskaper jämfört med deras experimentella motsvarighet. Även de vedertagna idéerna att använda sig av ett tröskelvärde och att generera EPSP som en Poissonprocess visade sig fungera som förväntat. Vi använder oss dock inte enbart av enkla funktioner för att återskapa olika komponenters utseende. Det är främst funktionen som genererar EPSP som är krävande då den inte direkt använder sig av parametrar som hämtats från mätningarna. Programmeringsalgoritmen tar alltså lång tid att köras.

Mer tillgänglig mätdata hade hjälpt att få mer information till de statistiska undersökningarna, och även chans att upptäcka beteenden som inte har hittats hittills. Dessutom kan forstatt utveckling av metoder som används för att identifiera stor-

heter, bidra till en förbättring av modellen. Detaljrikedomen i den experimentella datan, och den stora variationen av beteende som återfinns i den, tyder på att mycket arbete skulle kunna fortsätta spenderas på att fånga dess karaktäristik till modellen. Samtidigt kan det vara värt att tänka på balansen mellan en modell som är detaljerad och en modell som är lättanvänd.

Under arbetets gång har vi fått mer insikt och kunskap inom främst statistik, modellering, datahantering, och neurovetenskap. Utvecklingen av modellen har varit en intressant process och förhoppningsvis ska vår undersökning kunna ge insikt till modelleringsområdet, demonstrera hur den här typen av modell kan anpassas till mätningar, och kanske också vara intressant för forskare inom området. Vad gäller vidare forskning om neuroner så önskas fortsatta framsteg för förståelsen av hjärnans många delar och egenskaper.

Referenser

[1] Centre for Brain Research (Indian Institute of science). URL: http://www.cns.iisc.ernet.in/cbr/ (hämtad 2015-03-13).

[2] The Human Brain Project. URL: www.humanbrainproject.eu/faq/general (hämtad 2015-03-05).

[3] Alzheimersfonden. URL: http://www.alzheimerfonden.se/om_demens/alzheimers_sjukdom/mer_om_alzheimers_sjukdom (hämtad 2015-03-06).

[4] S. Sartov Jensen, O Naesh och H Askmark, "Parkinsons sjukdom - en utmaning för anestesiologen", *Läkartidningen*, nr 23, s. 1552–1555, 8 juni 2010. URL: http://www.lakartidningen.se/Functions/OldArticleView.aspx?articleId=14519 (hämtad 2015-03-04).

[5] K Andersen, "Enthusiasts and skeptics debate artificial intelligence", *Vanity Fair*, 26 nov. 2014. URL: http://www.vanityfair.com/news/tech/2014/11/artificial-intelligence-singularity-theory (hämtad 2015-03-06).

[6] X.-J. Wang, "Neurophysiological and computational principles of cortical rythms in cognition", *Physiology Reviews*, vol. 90, nr 23, s. 1195–1268, 2010. DOI: 10.1152/physrev.00035.2008.

[7] J. Feng, "Is the integrate-and-fire model good enough? a review", *Neural Networks*, vol. 14, nr 6-7, s. 955–, 2001. DOI: 10.1016/S0893-6080(01)00074-0.

[8] A. Hodgkin och A. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *The Journal of Physiology*, vol. 117, nr 4, s. 500–544, 28 aug. 1952. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/?page=1 (hämtad 2015-05-08).

[9] L. Sacerdote och M. T. Giraudo, "Stochastic integrate and fire models: A review of mathematical methods and their applications", i *Stochastic Biomathematical Models - With Applications to Neuronal Modeling*, M. Bachar och J. Batzel, utg. Berlin: Springer, 2013, vol. 2058, s. 99–148.

[10] R. Jolivet, A. Roth, F. Shurmann, W. Gerstner och W. Senn, "Special issue on quantitative neuron modeling", *Biological Cybernetics*, vol. 99, nr 4-5, s. 237–239, 5 nov. 2008. DOI: 10.1007/s00422-008-0274-5.

[11] R. Thompson, *The Brain: A Neuroscience Primer*, 3. utg. New York: Worth Publishers, 2000.

[12] N. Rougier, *Neuron figure no text (modifierad, med tillagd text i figur)*. URL: http://commons.wikimedia.org/wiki/File\%3ANeuron-figure-notext.svg (hämtad 2015-06-04).

[13] J. M. Lackie, utg., *The Dictionary of Cell & Molecular Biology*, 5. utg. Amsterdam: Elsevier, 2013.

[14] A. Mason, A. Nicoll och K. Stratford, "Synaptic transmission between individual pyramidal neurons of the rat the visual cortex in vitro", *The Journal of Neuroscience*, vol. 11, nr 1, s. 72–84, jan. 1991. URL: http://www.jneurosci.org/content/11/1/72.full.pdf+html (hämtad 2015-05-19).

[15] A. Fletcher, "Action potential: Generation and propagation", *Anaesthesia & Intensive Care Medicine*, vol. 12, nr 6, s. 258–262, juni 2011. DOI: 10.1016/j.mpaic.2011.03.010.

[16] M. D. Binder, N. Hirokawa och U. Windhorst, utg., *Encyclopedia of Neuroscience*. New York: Springer, 2009, vol. 1.

[17] A. Burkitt, "A review of the integrate- and fire model: I. homogenous synaptic input", *Biological Cybernetics*, vol. 95, nr 1, s. 1–19, juli 2006. DOI: 10.1007/s00422-006-0068-6.

[18] D. Horn och I. Opher, "Complex dynamics of neuronal thresholds", *Neurocomputing*, vol. 3233, nr 0, s. 161 –166, 2000, ISSN: 0925-2312. DOI: http://dx.doi.org/10.1016/S0925-2312(00)00158-2. URL: http://www.sciencedirect.com/science/article/pii/S0925231200001582.

[19] B. P. Bean, "The action potential in mammalian central neurons", *Nature Reviews Neuroscience*, vol. 8, nr 6, s. 451–465, juni 2007. DOI: 10.1038/nrn2148.

[20] G. Maimon och J. A. Assad, "Beyond poisson: Increased spike-time regularity across primate parietal cortex", *Neuron*, vol. 62, nr 3, s. 426–440, 14 maj 2009. DOI: 10.1016/j.neuron.2009.03.021.

[21] Y. Yarom och J. Hounsgaard, "Voltage fluctuations in neurons: Signal or noise?", *Physiological Reviews*, vol. 91, nr 3, 1 juli 2011. DOI: 10.1152/physrev.00019.2010.

[22] A. Bjorefeldt, U. Andreasson, J. Daborg, I. Riebe, P. Wasling, H. Zetterberg och E. Hanse, "Human cerebrospinal fluid increases the excitability of pyramidal neurons in the in vitro brain slice", *The Journal of Physiology*, vol. 593, nr 1, s. 241–243, jan. 2015. DOI: 10.1113/jphysiol.2014.284711.

[23] J. M. Bekkers, "Pyramidal neurons", *Current Biology*, vol. 21, nr 24, R975, 20 dec. 2011. DOI: 10.1016/j.cub.2011.10.037.

[24] J. A. Rice, *Mathematical Statistics and Data Analysis, International Edition*, 3. utg. Belmont: Brooks/Cole Cengage Learning, 2013.

[25] K. I. Najim och A.-K E. Daoud, *Stochastic Processes: Estimation, Optimization and Analysis*, 1. utg. London och Sterling VA: Kogan Pager Science, 2004.

[26] R. M. Feldman och C. Valdez-Flores, *Applied Probability and Stochastic Processes*, 2. utg. Berlin Heidelberg: Springer, 2010.

[27] W. G. Cochran, "The χ^2 test of goodness of fit", *The Annals of Mathematical Statistics*, vol. 23, nr 3, s. 315–345, 1953. URL: http://projecteuclid.org/euclid.aoms/1177729380 (hämtad 2015-04-28).

[28] Okänd. Föreläsningsanteckningar. Ämne: "χ^2-tests for independance, Goodness-of-fit." STA114, Duke University, 2011. URL: https://stat.duke.edu/courses/Fall11/sta114/chisq.pdf (hämtad 2015-05-02).

[29] G. C. Blain, "Revisiting the critical values of the lilliefors test: Towards the correct agrometeorological use of the kolmogorov-smirnov framework", *Bragantia*, vol. 73, nr 2, 10 juni 2014. URL: http://dx.doi.org/10.1590/brag.2014.015 (hämtad 2015-05-02).

[30] Mathworks, *Evaluating goodness of fit*. URL: http://se.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html (hämtad 2015-05-18).

[31] ——, *Coefficient of determination (r-squared)*. URL: http://se.mathworks.com/help/stats/coefficient-of-determination-r-squared.html (hämtad 2015-05-09).

[32] TempUnit, *Epsp on a point neuron*. URL: http://www.tempunit.org/?tag=mathematical-function (hämtad 2015-03-06).

A

Maximum Likelihoodmetoden

För att anpassa mätdata efter sannolikhetsfördelningar krävs en metod för att uppskatta de parametrar som beskriver fördelningen. Den metod som täcks här kallas för Maximum Likelihoodmetoden såsom presenterad av Rice [24].

Antag att datan kan modelleras som en stokastisk variabel X, och att den består av N mätpunkter $x_1, \ldots, x_N$. Ansätt sedan en sannolikhetsfördelning med M parametrar $\theta_1, \ldots, \theta_M$. Då skrivs den så kallade likelihoodfunktionen som

$$L(x_1, \ldots, x_N; \theta_1, \ldots, \theta_M) = P(x_1, \ldots, x_N; \theta_1, \ldots, \theta_M). \tag{A.1}$$

Metoden går då ut på att maximera L, vilket görs genom att lösa ekvationssystemet

$$\begin{cases} \frac{\partial L}{\partial \theta_1} = 0 \\ \vdots \\ \frac{\partial L}{\partial \theta_M} = 0 \end{cases}. \tag{A.2}$$

Detta görs enklare om $x_1, \ldots, x_N$ är oberoende. I så fall kan L skrivas som

$$L(x_1, \ldots, x_N; \theta_1, \ldots, \theta_M) = \prod_{n=1}^{N} P(x_n; \theta_1, \ldots, \theta_M). \tag{A.3}$$

För att förenkla metoden ytterligare kan det utnyttjas att $L > 0$ samt att logaritmfunktionen är monotont växande för att slutleda att de $\theta_1, \ldots, \theta_M$ som maximerar L även maximerar $\log(L)$. Alltså räcker det med att lösa ekvationssystemet för

$$\log(L(x_1, \ldots, x_N; \theta_1, \ldots, \theta_M)) = \sum_{n=1}^{N} \log(P(x_n; \theta_1, \ldots, \theta_M)). \tag{A.4}$$

B

Data för brusmodellen

Tabell B.1: Tabell över de frekvenskomponenter med motsvarande amplitud som användes för att generera bruset genom en summa av sinustermer.

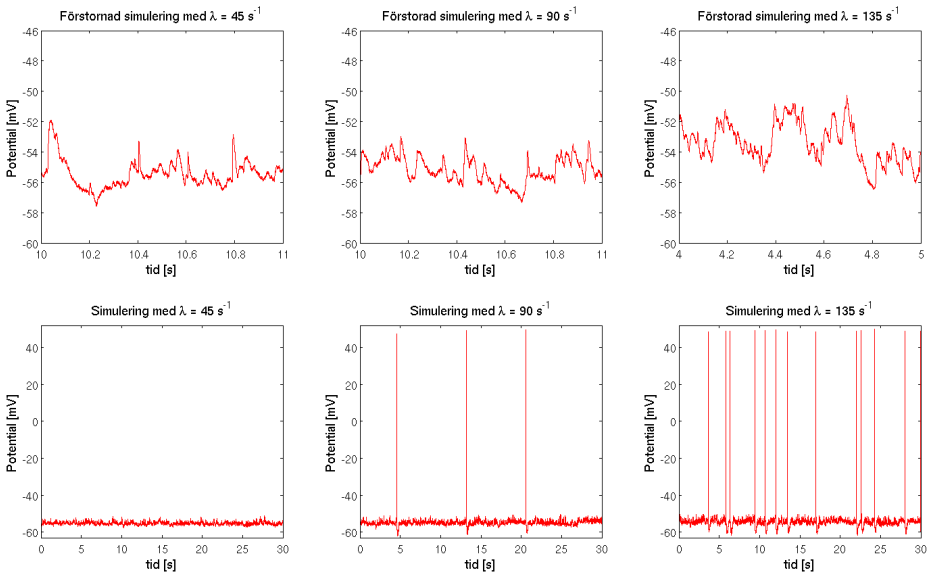
Frekvens [Rad/s]	Amplitud $\cdot 2 \times 10^{-6}$ [V]	Frekvens [Rad/s]	Amplitud $\cdot 2 \times 10^{-6}$ [V]	Frekvens [Rad/s]	Amplitud $\cdot 2 \times 10^{-6}$ [V]
0,1047	100	39,9	50	1186	5
0,5236	97	45	48	1256	5
0,6283	95	50	46	1570	5
0,733	92	6	44	1685	5
0,8378	91	72,1	42	1885	5
1,361	90	82,52	40	1918	5
1,571	85	92,5	38	2079	5
2,094	82	98,5	36	2362	5
2,409	80	109	34	3692	5
3,336	78	120	32	4089	5
3,665	77	144,2	30	4136	5
5,445	76	155	28	4579	5
5,969	74	226	26	5178	5
8,692	72	256	24	5280	5
9,32	70	314,2	22	6353	5
10,58	68	377,1	20	7240	5
13,09	66	410,2	18	7395	5
16,44	64	521,1	14	7684	5
17,17	62	622,45	12	7840	5
17,7	60	630	10	8570	5
21,26	58	742,8	5	9901	5
25,13	56	942	5	11670	5
26,28	54	1031	5	12120	5
31,21	52	1075	5	12570	10

C

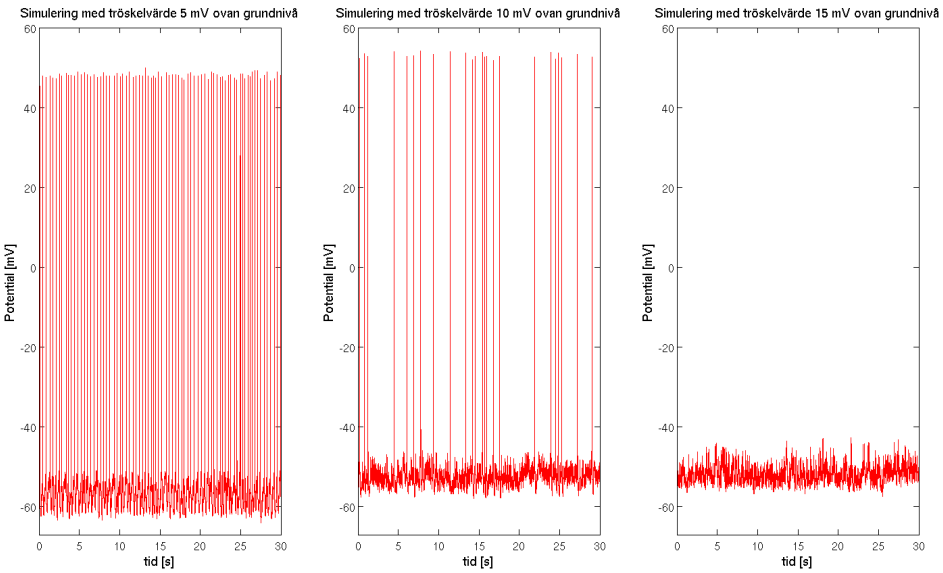
Exempel på egenskaper hos simulering och mätdata

C.1 Simuleringar med variation av parametrar

I denna bilaga presenteras figurer där olika parametrar i modellen varierats.

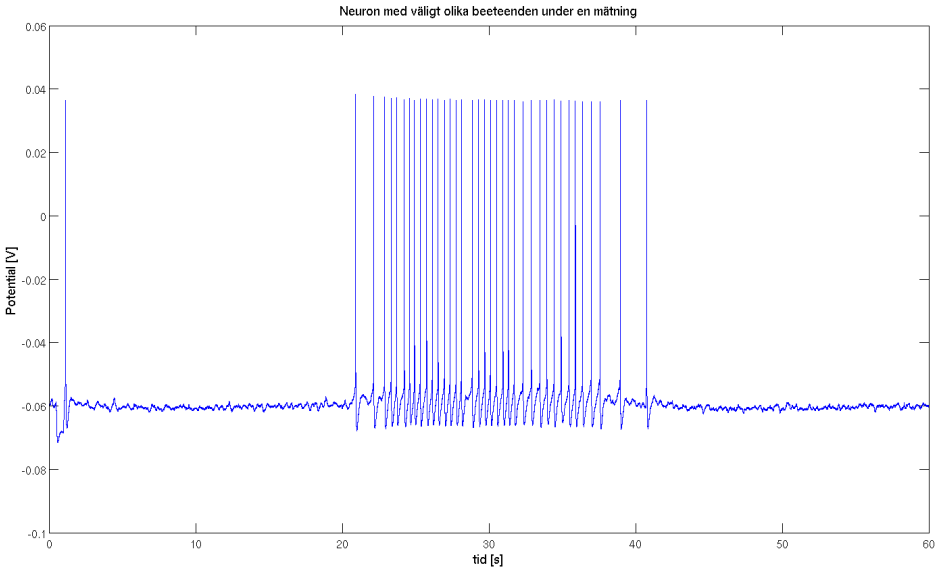


Figur C.1: Figuren visar effekten av att variera intensitetsparametern för EPSP-generering ±50%. Intensitetsparametern ökar från vänster till höger. I figurerna ökar spikfrekvens med ökad intensitetsparameter. De tre övre figurerna visar en förstoring av simuleringen där variation EPSP-frekvens kan observeras, främst genom att enstaka EPSP blir svåra att urskilja i den mätning med en 50-procentig ökning av intensitetsparametern.

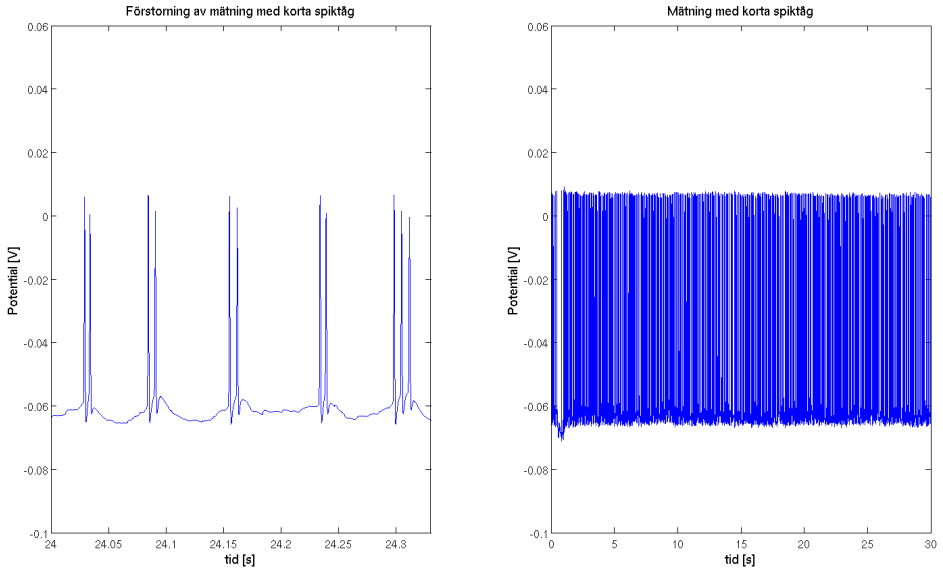


Figur C.2: I figuren har tröskelvärdet varierats i förhållande till grundnivån medan alla andra parametrar hållits konstanta. Figuren i mitten har ett tröskelvärde likt medelvärdet för alla mätningar som undersökts. Den vänstra figuren har ett tröskelvärde som ligger 5 mV ovan grundnivån, vilket är 50% mindre än tröskelvärdet av den mellersta figuren och det är synligt hur spikfrekvensen ökar då tröskelvärdet minskas. I figuren till höger har tröskelvärdet istället ökats med 50% och inga spikar genereras. Det framgår även i figuren att spikarnas amplitud ökar med ökat tröskelvärde då deras amplitud definierats med avseende på tröskelvärdet och inte grundnivån.

C.2 Figurer av mätningar på neuroner med ovanliga beeteenden



Figur C.3: I mätningen i figuren ovan uppvisar neuronen en period av låg EPSP-frekvens, och också utan några spikar, och sedan en period av högre spikfrekvens. Beteendet skulle kunna bero på en varierande intensitetsparameter eller variation i tröskelvärde, och skulle kanske kunna genereras om sådana variationer infördes i modellen.

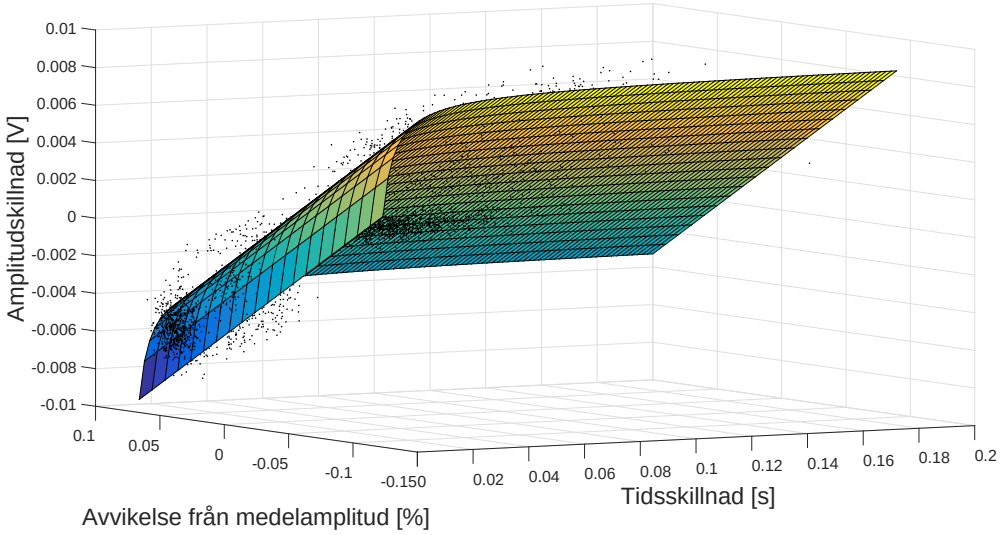


Figur C.4: Figuren till vänster visar ett mönster av korta spiktåg. Beteendet kan grunda sig i natriumströmmar som ökar potentialen även efter potentialen återgått till grundnivån efter en spik, en ökning som i sin tur kan generera en ny spik. Till höger visas samma mätning men inte förstorad för att visa hur tätt dessa spiktåg kommer.

D

Funktionsanpassningar

D.1 Funktion för spikamplitud



Figur D.1: Figuren visar den funktionsanpassning som gjorts för spikarnas amplitudvariering. På korta tidsavstånd beter sig funktionen som en potensfunktion och för större avstånd som ett plan.

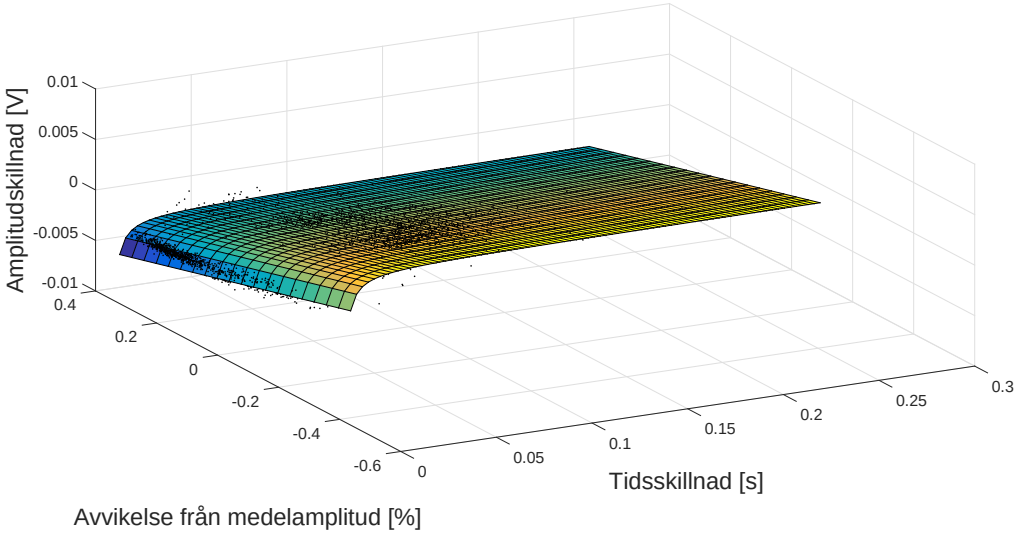
```
Resultatet från Curve Fitting Tool i MATLAB:

General model:
     f(x,y) = a/x+b*y+c
Coefficients (with 95% confidence bounds):
       a =  -2.769e-05  (-2.804e-05, -2.733e-05)
       b =    -0.06113  (-0.06178, -0.06049)
       c =     0.00176  (0.001729, 0.001791)

Goodness of fit:
  SSE: 0.01557
  R-square: 0.9109
  Adjusted R-square: 0.9109
  RMSE: 0.001159

x=tidsavstånd
y=procentuellt avstånd från medelvärdet
z=potentialskillnad mellan två närmaste spikar
```

D.2 Funktion för efterpotentialens amplitud



Figur D.2: Figuren visar den funktionsanpassning som gjorts för efterpotentialernas amplitudvariering. På korta tidsavstånd beter sig funktionen som en potensfunktion och för större avstånd som ett plan.

```
Resultatet från Curve Fitting Tool i MATLAB:

General model:
     f(x,y) = a/x+b*y+c
Coefficients (with 95% confidence bounds):
       a =  -1.776e-05  (-1.799e-05, -1.753e-05)
       b =   -0.008532  (-0.008635, -0.008428)
       c =    0.001114  (0.001093, 0.001134)

Goodness of fit:
  SSE: 0.007152
  R-square: 0.8731
  Adjusted R-square: 0.873
  RMSE: 0.0007852

 x=tidsavstånd
 y=procentuellt avstånd från medelvärdet
 z=potentiallskillnad mellan två närmaste efterpotentialer
```

E

MATLAB-kod

E.1 Simulering

```
%% Simulation of a neuron.
clc %Clear Command Window.

%For comparison with a measurement, load data.
load Data
load tres
load baseLine
load meanAmp
load meanDip
load fileNames


measurement = 1; %Set the measurement to compare with (1-18).
srate = 10000; %Set the sample rate (Hertz).
duration = 10; %Length of the simulation, usually between 0-60 (seconds
    ).

%Get data for the measurment to compare, needs specific files.
filepath = ['data files/' char(fileNames(measurement)) '.txt']; %Builds
     the file path
fid = fopen(filepath, 'r');
data = textscan(fid, '%f', 'headerLines', 1); %Skips the first line of
    text.
fclose(fid);
data = cell2mat(data); %Creates a data matrix
timeData = linspace(1/srate,length(data)/srate,length(data)); %Time
    vector adjusts to the measurement.


%%%%%%%%% Set 14 parameters %%%%%%%%%

%The potential at rest for the neuron (volt).
baseLine = baseLine(measurement);

%The potential over which a spike should be generated (volt).
treshold = findTreshold(data,tres(measurement));

%Parameters for EPSP-generation.
paramrise = [0.5*8.6074; 2*7.4521e-4];
```

```
paramamprise = [0.5*3.3017; 3.6723e-4];
Pfit = [0.5*2.1226e-1; 0.18478e-5 - 1.1133e-3];

%The intensity of EPSP.
lambda = 200; %(1/second)

%Parameters for spike-generation.
meanSpikeAmplitude = meanAmp(measurement); %Mean amplitude for spikes, relative treshold (volt).
meanDipAmplitude = meanDip(measurement); %Mean amplitude for dips, positive relative treshold (volt).
spikeRiseTime = mean(getData('risetime',measurement)); %Risetime for spikes, treshold to spike-maximum (seconds).
spikeFallTime = mean(getData('falltime',measurement)); %Falltime for spikes, maximum to treshold (seconds).
dipFallTime = mean(getData('falltimedip',measurement)); %Falltime for dips, treshold to dip-maximum (seconds).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Set a random value where the noiseTime will start and create a vector long enough (2 min) with those times,
noiseTimeStart = round(srate*60*rand)/srate;
noiseTime = linspace(noiseTimeStart+1/srate,noiseTimeStart+120,2*length(data));


%Time- and potential-vector for the simulation.
time = linspace(1/srate,duration,srate*duration); %Generates a time vector for the simulation.
potential = repmat(baseLine,1,length(time)); %Create a potential vector with baseline value.


%%%% Set which times EPSP will start %%%%

%Generates n exponentially distributed parameters with the intensity-parameter lambda.
n = 1000000;
X = -log(rand(n,1))/lambda;

T = X(1);
i = 0;
EPSPLocations = []; %Clear the vector.
while T <= max(time);
    EPSPLocations(i+1) = T; %Vector of times when a EPSP should be generated.
    i = i + 1;
    T = T + X(i+1);
end

if isvector(EPSPLocations) %Check if vector.
    nextEPSP = EPSPLocations(1); %Time when next EPSP should be generated.
    nextEPSPCount = 1; %Index of EPSPLocations.
end
```

```
81
%Initial-values for spike-generation.
83 oldTime = 0;
oldSpikeAmplitude = meanSpikeAmplitude;
85 oldDipAmplitude = meanDipAmplitude;

87 i = 1; %Index for the whole simulation, increase for each time-step.
noiseIndex = 1; %Index for the noise-generation.
89 generationTime=[]; %Clear vector.

91
%%%%%%%%%%%%%%%%%%%%%%%%% The Simulation %%%%%%%%%%%%%%%%%%%%%%%%%
93

95 while i <= length(time)
    potential(i) = potential(i) + generateNoise(noiseTime(noiseIndex));
    %Add noise for each time-step.
97
    if potential(i) >= treshold %Over treshold?
99      newTime = i/srate; %Time when the spike starts.
        dt = newTime - oldTime; %Time between spikes.
101     newSpikeAmplitude = getSpikeAmplitude(dt,oldSpikeAmplitude,
    meanSpikeAmplitude); %Set the spike amplitude.
        newDipAmplitude = getDipAmplitude(dt,oldDipAmplitude,
    meanDipAmplitude); %Set the dip amplitude.
103
        %Create the spike.
105     [spikeTime,spikePotential,k] = generateSpike(treshold,baseLine,
    newSpikeAmplitude,newDipAmplitude,spikeRiseTime,spikeFallTime,
    dipFallTime,srate);

107     %Overwrithe old values.
        oldSpikeAmplitude = newSpikeAmplitude;
109     oldDipAmplitude = newDipAmplitude;
        oldTime = newTime;
111
        %Plot the spike.
113     for j = 1:length(spikeTime)
            if i+j <= length(time)
115         potential(i+j) = spikePotential(j);
            end
117     end

119     i = i + k; %Nothing can happen until the dip has been plotted.
    Skip loop until it is done.

121     %Skip EPSP during spike-generation.
        if i <= length(time) && nextEPSPCount <= length(EPSPLocations)
123         while EPSPLocations(nextEPSPCount) < time(i) &&
    nextEPSPCount <= length(EPSPLocations)
            nextEPSPCount = nextEPSPCount + 1;
125         end
            nextEPSP = EPSPLocations(nextEPSPCount);
127         potential(i-k+j:end) = baseLine;
        end
```

```
129
        %Make less discontinuous between the dip and the noise.
131     %Step forward with noiseTime to make noise approx = 0.
        while (sign(generateNoise(noiseTime(noiseIndex))) - sign(
    generateNoise(noiseTime(noiseIndex+1)))) == 0
133         noiseIndex = noiseIndex + 1;
        end
135
    end
137

139 %Time to generate EPSP.
    if i <= length(time) && isvector(EPSPLocations) && time(i) >=
    nextEPSP && nextEPSPCount <= length(EPSPLocations)
141     tic;
        %Create EPSP.
143     [EPSPTime,EPSPPotential] = generateEPSP(paramrise, paramamprise
    , Pfit, srate);
        generationTime(nextEPSPCount)=toc;
145
        nextEPSPCount = nextEPSPCount + 1; %Increase counter.
147     if nextEPSPCount <= length(EPSPLocations)
            nextEPSP = EPSPLocations(nextEPSPCount); %Get next time
    EPSP should be generated.
149     end

151     clc
        %Display process of simulation and remaining time.
153     disp(['Generating: ' num2str(nextEPSPCount/length(EPSPLocations
    )*100) '%']);
        meanTime=mean(generationTime);
155     disp(['Time to completion: ' num2str(round((length(
    EPSPLocations)+1-nextEPSPCount)*meanTime)) ' seconds']);

157     %Plot EPSP.
        for j = 1:length(EPSPTime)
159         if i+j <= length(time)
            potential(i+j) = potential(i+j) + EPSPPotential(j);
161         end
        end
163
    end
165
    i = i + 1; %Increase time-index.
167 noiseIndex = noiseIndex + 1; %Increase noise-index.

169 end

171
potential = potential(1:length(time)); %Removes end of potential if
    vector is too long.
173

175 clf
plot(time,potential*1000,'r') %Plot simulation.
177 hold on
```

```
plot(timeData,data*1000) %Plot data to compare with, if so wish.
axis([0 60 -70 60])

%Set label
xlhand = get(gca,'xlabel');
set(xlhand,'string','Tid [s]','fontsize',30)
ylhand = get(gca,'ylabel');
set(ylhand,'string','Potential [mV]','fontsize',30)
set(gca,'fontsize',15)


%Plot baseline and treshold, if so wish.
plot([0,timeData(end)],1000*[treshold treshold], 'g-')
plot([0,timeData(end)],1000*[baseLine baseLine], 'g-')

shg
```

Listing E.1: Simulering

E.2 Hitta tröskelvärde

```
%%%% Return treshold %%%%

function [tresh] = findTreshold(V,startTresh)
% Returns a treshold. Requires a vector with voltage from a measurement
    and
% an estimate of the treshold. The estimate must be larger than the
   actual
% treshold. The estimate is required to exclude spikes.

    tresh=-100; %This will become the treshold

    %Returns NaN if estimate is NaN.
    if isnan(startTresh)
        tresh=NaN;
        return
    end

    %Checks all values in the voltage vector. If the value is a local
    %maximum, above the tresh vector and below the estimate, the value
    will
    %be saved to the tresh vector.
    for i=2:length(V)-1
        if V(i) > V(i-1) && V(i) > V(i+1) && V(i)<startTresh && V(i)>
    tresh
            tresh=V(i);
        end
    end

    tresh=tresh+0.00001; %Adds some margin
end
```

Listing E.2: Hitta tröskelvärde

E.3 Spikgenerering

```
%%%% Function to generate the form of a spike and dip %%%%

%Part 1: the rising of the spike from treshold to maximum.
%Part 2: the falling of the spike from maximum to treshold.
%Part 3: the falling of the dip and rising to baseline. Treshold to dip
    -maximum to baseline.

function [time,potential,n] = generateSpike(treshold,baseline,
    spikeAmplitude,dipAmplitude,riseTime,fallTime,dipFallTime,srate)

noiseStart = round(srate*60*rand)/srate; %Random start of noiseTime.

%Amplitudes of the three parts relative treshold.
Aspike1 = spikeAmplitude;
Aspike2 = spikeAmplitude+dipAmplitude;
Adip = dipAmplitude;

%Setting the gamma-parameters.
gammaSpike1 = riseTime/3;
gammaSpike2 = fallTime/3;

%Expand to have marginals.
riseTimeDip = dipFallTime*3;
fallTime = fallTime*5;

%Time for spike-maximum.
t0 = riseTime;


%Lorenztian-functions for the spike.
V1 = @(t) Aspike1*gammaSpike1.^2./((t-t0).^2+(gammaSpike1).^2);
V2 = @(t) Aspike2*gammaSpike2.^2./((t-t0).^2+(gammaSpike2).^2);


%Time and potential for part 1.
trise = linspace(1/srate,riseTime,round(riseTime*srate));
potentialSpike1 = V1(trise) + generateNoise(noiseStart+trise);


%Time and potential for part 2.
tfall = linspace(riseTime+1/srate,riseTime+fallTime,round(fallTime*
    srate));
potentialSpike2 = V2(tfall) - Adip + generateNoise(noiseStart+trise(end
    )+tfall);
potentialSpike2(potentialSpike2<0) = []; %Cut time when under treshold.
tfall = tfall(1:length(potentialSpike2)); %Adjust time.


%Function for dip.
T = dipFallTime;
K = Adip*exp(1)/T;
V3 = @(t) -K*t.*exp(-t./T);
```

```
%Time and potential for part 3.
tfallAndRise = linspace(0,dipFallTime+riseTimeDip,round((dipFallTime+
    riseTimeDip)*srate)+1);
potentialDip = V3(tfallAndRise) + generateNoise(noiseStart+trise(end)+
    tfall(end)+tfallAndRise);


%Cut potential over baseline.
i = 1;
while i <= length(potentialDip)
    if tfallAndRise(i) > T && potentialDip(i) > (baseline-treshold)
        potentialDip(i) = [];
    else
        i = i + 1;
    end
end
tfallAndRise = tfallAndRise(1:length(potentialDip)) + tfall(end); %
    Adjust time.

%Total time-vector.
time = [trise tfall tfallAndRise];
%Total potential-vector adjust by treshold.
potential = [potentialSpike1 potentialSpike2 potentialDip] + treshold;

%Index for which the dip has reached it's maximum.
%Add n to time-index in simulation.
n = length([trise tfall]) + round(dipFallTime*srate);
end
```

Listing E.3: Spikgenerering

E.4 Beräkna amplitud för spik

```
%%%%% Function to set the amplitude of a spike %%%%%

function [newAmp] = getSpikeAmplitude(dt,oldAmp,meanAmp)
    %Constants from cftool
    a = -2.769e-5;
    b = -0.06113;
    c = 0.00176;

    f = @(x,y) a/x+b*y+c;

    X = dt;
    Y = (oldAmp-meanAmp)/meanAmp;
    Z = f(X,Y);

    dAmp = Z;

    newAmp = oldAmp + dAmp;
end
```

Listing E.4: Amplitudberäkning för spik

E.5 Beräkna amplitud för efterpotential

```
%%%% Function to set the amplitude of a dip %%%%

function [newAmp] = getDipAmplitude(dt,oldAmp,meanAmp)
    %Constants from cftool
    a = -1.776e-5;
    b = -0.008532;
    c = 0.001114;

    f = @(x,y) a/x+b*y+c;

    X = dt;
    Y = (oldAmp-meanAmp)/meanAmp;
    Z = f(X,Y);

    dAmp = Z;

    newAmp = oldAmp + dAmp;

end
```

Listing E.5: Amplitudberäkning för efterpotential

E.6 Generering av EPSP

```
%%%% A function that generates a single EPSP %%%%

function [time, potential] = generateEPSP(paramRise, paramAmprise, Pfit
    , srate)

%Generates a gamma distributed rise time.
tr = gamrnd(paramRise(1), paramRise(2));

%Generates the amplitude as a linear function of the risetime plus a
    gamma
%distributed stochastic variable. Also ensures that the amplitude is
%positive (a negative amplitude can be the result of poorly chosen
%parameters).
A = -1;
while A <= 0
    A = Pfit(1)*tr + Pfit(2) + gamrnd(paramAmprise(1), paramAmprise(2))
    ;
end

%Solves a nonlinear system of equations to get the parameters tau and
    taur
%from the rise time and amplitude.
f = @(x)eqsyst(x, tr);
guess = [0.01; 0.006];
options = optimset('Display','off');
X = fsolve(f, guess, options);
```

```
tau = X(1);
taur = X(2);

%Calculates the w parameter.
w = A*(X(1) - X(2))./((X(2)./X(1)).^(X(2)./(X(1) - X(2))) - (X(2)./X(1)
    ).^(X(1)./(X(1) - X(2))));

%Defines a function handle V(t) representing the potential of the EPSP
    as a
%function of time.
V = @(t) w/(tau - taur).*(exp(-t./tau) - exp(-t./taur));

%Constructs time and potential vectors describing the EPSP until 99 %
    of it
%has passed.
i=1;
while true
    time(i)=1/srate*i;
potential(i)=V(time(i));
    if potential(i) < 0.01*A && time(i) > tr
        return
    end
i=i+1;
end
```

Listing E.6: Generering av EPSP

E.7 EPSP-ekvationssystem

```
%%%%% This function describes the equation system that has to be %%%%%
%%%%% solved each time an EPSP is generated                      %%%%%

function F = eqsyst(x, tr)

    F = [x(2).*x(1)./(x(2) - x(1)).*log(x(2)./x(1)) - tr
         exp(-tr./x(1)) - exp(-tr./x(2)) - ((x(2)./x(1)).^(x(2)./(x(1)
    - x(2))) - (x(2)./x(1)).^(x(1)./(x(1) - x(2))))];
end
```

Listing E.7: EPSP-ekvationssystem

E.8 Brusgenerering

```
%%%% Noise-generation by a fourier approximation %%%%

function [potentialChange] = generateNoise(time)

potentialChange=(100*sin(0.1047*time)+97*sin(0.5236*time)+95*sin
    (0.6283*time)+92*sin(0.733*time)+91*sin(0.8378*time)+ ...
90*sin(1.361*time)+85*sin(1.571*time)+82*sin(2.094*time)+80*sin(2.409*
    time)+78*sin(3.336*time)+ ...
77*sin(3.665*time)+76*sin(5.445*time)+74*sin(5.969*time)+72*sin(8.692*
    time)+70*sin(9.32*time)+ ...
68*sin(10.58*time)+66*sin(13.09*time)+64*sin(16.44*time)+62*sin(17.17*
    time)+60*sin(17.7*time)+ ...
58*sin(21.26*time)+56*sin(25.13*time)+54*sin(26.28*time)+52*sin(31.21*
    time)+50*sin(39.9*time)+ ...
48*sin(45*time)+46*sin(50*time)+44*sin(51.21*time)+42*sin(72.1*time)
    +40*sin(82.52*time)+ ...
38*sin(92.5*time)+36*sin(98.5*time)+34*sin(109*time)+32*sin(120*time)
    +30*sin(144.2*time)+ ...
28*sin(155*time)+26*sin(226*time)+24*sin(256*time)+22*sin(314.2*time)
    +20*sin(377.1*time)+ ...
18*sin(410.2*time)+14*sin(521.1*time)+12*sin(622.45*time)+10*sin(630*
    time)+5*(sin(742.8*time)+ ...
sin(942*time)+sin(1031*time)+sin(1075*time)+sin(1186*time)+sin(1256*
    time)+ ...
sin(1570*time)+sin(1685*time)+sin(1885*time)+sin(1918*time)+sin(2079*
    time)+ ...
sin(2362*time)+sin(3692*time)+sin(4089*time)+sin(4136*time)+sin(4579*
    time)+ ...
sin(5178*time)+sin(5280*time)+sin(6353*time)+sin(7240*time)+sin(7395*
    time)+ ...
sin(7684*time)+sin(7840*time)+sin(8570*time)+sin(9901*time)+sin(11670*
    time)+ ...
sin(12120*time))+10*sin(12570*time));

disturb = []; %Clear vector.
for i=1:length(potentialChange)
    disturb(i) = 10*rand*(2*rand-1); %Create random disturbance.
end
potentialChange = (potentialChange+disturb)*2e-6; %Add disturbance and
    scale down.

end
```

Listing E.8: Brusgenerering