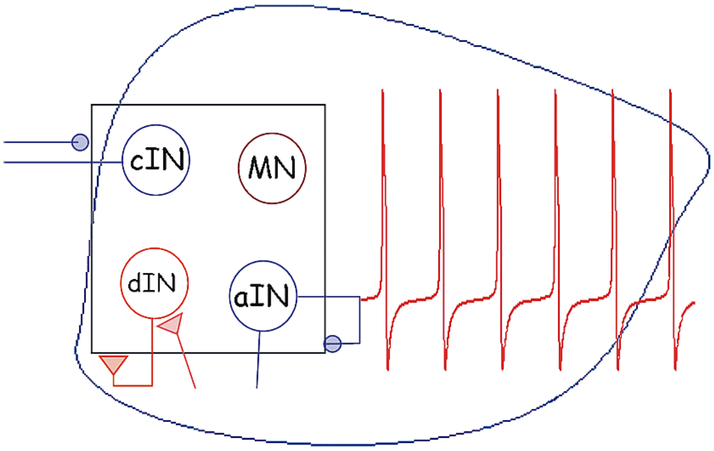
Dynamical Systems and their Applications in Neuroscience

Bart Sautois



Promotor: Prof. Dr. W. Govaerts

Copromotor: Dr. Yu.A. Kuznetsov, Universiteit Utrecht

Proefschrift ingediend tot het behalen van de graad van
Doctor in de Wetenschappen: Informatica

Academiejaar 2006–2007

Faculteit
Wetenschappen

Vakgroep Toegepaste Wiskunde
en Informatica

UNIVERSITEIT
GENT

First of all, I owe a great deal of thanks to my PhD supervisor Willy Govaerts. It is thanks to his guidance and support that this work can be presented here. All through these past 4 years, and even before that, his support and advice meant a lot to me.

I also want to thank all the other professors I have had the joy of collaborating with. Primarily this is Yuri Kuznetsov from Utrecht University, whose ideas and support were crucial in the development of MATCONT. And of course Alan Roberts and Stephen R. Soffe from the University of Bristol, thanks to whom I was able to apply my theoretical knowledge of neural modelling to a real-life biological project. My collaboration with them has been very interesting and satisfying, although it has also taught me the reality about availability of detailed biological data.

Of course I also owe a lot to my wife Sofie, who has supported me in my academic endeavours for many years, even before this PhD even started. I hereby thank her for listening to my ranting when things didn’t work out as I planned, and my cries of joy when they finally did.

I also thank her and my brother Tim for reading over my PhD. Any corrected mistakes are thanks to them, any remaining mistakes are my fault.

Finally I want to thank my parents, who have supported me throughout all of my studies, and everyone else who has contributed to my PhD either directly or indirectly.

Thank you very much!

Bart Sautois
November 2006

This research was supported by the Fund for Scientific Research - Flanders (Belgium) (F.W.O.-Vlaanderen).

Contents

Introduction

The most exciting phrase to hear in science, the one that heralds new discoveries, is not ’Eureka!’ (I found it!) but ’That’s funny ...’ (Isaac Asimov (1920 - 1992))

In dynamical systems, the evolution of a system is given by its present state and by rules defining its evolution in the future. Following these rules or equations, one can track the system for as long as required. This very general definition allows scientists from a wide variety of fields to use dynamical systems in their endeavours. Indeed, dynamical systems can be used to model any process for which one can define a finite set of rules to describe its evolution. Some of the many domains of modelling applications using dynamical systems are physics, finance, geology, biology and chemistry.

Because of the many possible applications, there is much interest in the general theory behind dynamical systems. The change of a system under variation of one parameter can explain many phenomena observed in nature and science. A radical change in a system’s behaviour at some parameter value is called a bifurcation; the study of these bifurcations is a very important field in mathematical science. In Chapter 1, the reader is introduced to continuous dynamical systems, and the most important bifurcations are listed here. This chapter should contain enough information on dynamical systems theory to guide the reader through the rest of this thesis.

A major field of application of dynamical systems is computational neuroscience. Dynamical systems are very regularly used for modelling neurons or neural networks, in variable levels of biological detail. The first neural model based on individual ionic currents in and out of the cell was developed by Hodgkin and Huxley in 1952. Since then, a large number of models has been constructed, based on the ideas behind their original model. In Chapter 2, we first give an elaborate introduction to the background involving cell membranes, ionic currents, channels, etc., followed by listing the equations of four well-known neural models built-up of dynamical systems, which will be used as primary examples in later sections in this thesis.

It is a daunting task to keep track of all existing neural models, and therefore the study of these models is a field of research on its own. One important concept that can aid in this effort, is the classification of the developed models. When a means of classification would exist, in which each category has a particular set of properties, and in which there are clear criteria for models to belong to any of these categories, then the study of newly developed models would be greatly simplified. This would then mainly consist of determining which category the new model belongs to, after which you would immediately know several of the model’s most important properties and/or functionalities. Some preliminary categorizations already exist, but they are clearly insufficient in some respect. In Section 2.3, we elucidate the problem with some existing classifications, and suggest an improved version.

In this second chapter we also introduce the Phase Response Curve. This basically mathematical concept has very important applications in neural modelling. It is a curve that describes the effect of an input on a periodic orbit. Since a spiking neuron can be characterized as a periodic orbit, the phase response curve can be used to indicate the effect of an incoming (e.g. synaptic) input on a spiking neural model. We derive a new and efficient mathematical method to compute this phase response curve of a periodic orbit.

Since the study of dynamical systems and bifurcations raises widespread interest among scientific audiences, several software packages have been written over the past few decades to aid in this study. Important and widely used examples of these are AUTO and CONTENT. However, all packages have some important disadvantages. AUTO is fast and efficient, but it has no user interface and is, partly because of that, not very user-friendly. It takes quite some investment of a scientist to understand the software enough to do rigorous studies of the desired dynamical systems. Also, it does not provide more advanced features such as the computation of codimension 2 bifurcations and normal form coefficients. CONTENT is much more user-friendly, but it is hard to install, and development and monitoring was falling behind for a while. An important disadvantage of both packages is the platform-dependence. Both were developed for Unix / Linux machines, while most scientists to the present day prefer to work in the MS Windows environment. And while CONTENT also exists for Windows, this version is not easy to install.

In 2000, the development was initiated of MATCONT, a new software package for the study of dynamical systems and their bifurcations, intended to address and cure the biggest problems with the existing software. It was completely written in MATLAB, to have full platform-independence, and to have good extendibility of the package. The user interface was based on that of CONTENT, to keep its user-friendly features, and to make it easier for users to switch from one package to the other.

Some important contributions to this software package are introduced in Chapter 3. The functionalities were extended with the option of computing the phase response curve of a limit cycle and its derivative. The implementation is based on the mathematical algorithm described in Chapter 2, and on top of that, it makes very efficient use of the continuation aspect of the MATCONT software package.

The user can now also detect and continue homoclinic bifurcations. These curves of homoclinics can be started from a Bogdanov-Takens bifurcation or from a periodic solution with high period or from an equilibrium. Many codim 2 bifurcations can be detected on these homoclinic curves.

Finally, we have improved the speed of the software significantly by introducing C-code among the MATLAB-routines. By using this C-code wisely, we have been able to increase the speed of limit cycle continuation more than half, while preserving MATCONT's typical ease of installation and platform-independence. In a final section in this chapter, we note an important bug that we have discovered in MATLAB's most recently added sparse routines.

MATCONT was developed as a mathematically oriented software package, primarily intended for the theoretical study of dynamical systems. Currently, it is often used in teaching and mathematical research. However, the possible applications are far from restricted to those areas. MATCONT can help in any research that involves dynamical systems. The theoretical study of the systems involved can help clarify certain observed phenomena. That MATCONT has very interesting applications in e.g. neuroscience, is illustrated in Chapter 4.

A first logical use of the software is in the bifurcation study of an existing neural model. The Morris-Lecar model is a widely used model, but so far a complete bifurcation analysis of the system had not been made anywhere. This type of full bifurcation diagram can be important to know which behaviour can occur at which parameter values in the system. This way, one can explain certain observed network behaviour, or elicit some specific type of behaviour if needed. We have for the first time made such a complete bifurcation diagram of this renowned model.

We also introduce several interesting applications of phase response curves, which are now an integral part of MATCONT. We show how these curves can be used to determine the synchronizing and/or phase-locking abilities of neural networks, and how the connection delay plays a crucial role in these abilities. We also show some interesting phenomena in the interaction of phase response curves and limit cycle bifurcations.

Dynamical systems are not only useful in theoretical neuroscience. Also in applied neuroscience they have an important role to fulfill. When dynamical systems are used to model physiological neurons, the resulting models can be used to study the interaction of these neurons in a network, constructed using the available biological knowledge.

In collaboration with a group of biologists at the University of Bristol, we have built detailed models of the neurons in the spinal cord of the hatchling *Xenopus laevis*. The biological background of the animal and the equations and parameters used for the models of individual neurons and connections are given in Chapter 5.

In Chapter 6, we focus on the initial aim of this collaboration. The individual models of neurons and synapses are used to construct biologically realistic networks of neurons, as they are found in the young *Xenopus* spinal cord.

The first network we built was intended to simulate the swimming behaviour of the tadpole. The actual interactions in this network were already well-known, and this network was primarily used to verify our individual neuron and synapse models, but also to demonstrate the importance of specificity in neural modelling. We show that to disregard some important differences that we have created in the models for different neurons, will result in breakdown of the good network output.

Then we have used these identical individual model settings to study another hypothesis, regarding synaptogenesis. In the physiological network, most neurons are more likely to have synaptic connections with some specific types than with other ones. The hypothesis is that this specificity in connection is purely based on anatomical organization of the neurons, and not on the ability of growing synapses to make a distinction between the different types of neurons. The fact that one neuron would primarily connect to one or more specific types of neurons, would be purely caused by the dorso-ventral position of each of these types of neurons, and the place where the synapses actually grow. We have tested this hypothesis by building networks using the same swimming neurons as in the previous model, but now creating synapses on probabilistic grounds. The modelling results, which strongly support our hypothesis, are given in the second section of Chapter 6.

Finally, a network is currently being constructed to model the second type of tadpole behaviour, namely struggling. In this process, alternating bursts occur on both sides of the body (contrarily to swimming, which involves alternating single spikes). Of this network and behaviour, much less is known than of swimming. But our modelling has shown, among other things, that the most likely burst-terminating mechanism would be synaptic plasticity: by sending signals at high frequency over one synapse to another neuron, a synapse will weaken and in the long run it will have weakened to such extent that it will temporarily stop transmitting the signal, until it has sufficiently recovered. By incorporating this feature in our model, we were able to reproduce quite reliable alternating bursts of spikes in both sides of the *Xenopus*’ body.

The results in this thesis have been published in or submitted for publication by specialized scientific journals or proceedings, see [33], [49], [57], [58], [59], [60], [118], [35].

Chapter 1

Introduction to dynamical systems

This chapter introduces the reader to dynamical systems, listing the mathematical background needed for understanding the rest of this thesis. After the general basics of dynamical systems study, an elaboration is given on the different types of bifurcations, primarily equilibrium and limit cycle bifurcations, and finally homoclinic orbits are introduced.

1.1 Basics on ODEs

A general system of ordinary differential equations (ODEs) can be written as

$$\left\{ \begin{array}{rcl} \dfrac{dx_1}{dt} & = & f_1(x_1, x_2, \ldots, x_n), \\ \dfrac{dx_2}{dt} & = & f_2(x_1, x_2, \ldots, x_n), \\ & \vdots & \\ \dfrac{dx_n}{dt} & = & f_n(x_1, x_2, \ldots, x_n), \end{array} \right.$$

where each component f_i is assumed to have continuous mixed partial derivatives with respect to all x_j up to (and including) a sufficiently high order k. Introduce a state (phase) variable vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

and define a map $f : \mathbb{R}^n \to \mathbb{R}^n$ by

$$x \mapsto f(x) = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{pmatrix}.$$

We write $f \in C^k(\mathbb{R}^n, \mathbb{R}^n)$. Then the system can be written as one vector ODE

$$\dot{x} \equiv \frac{dx}{dt} = f(x), \quad x \in \mathbb{R}^n. \tag{1.1}$$

Denote by $\varphi^t(x_0)$ a solution of (1.1) that starts at $t = 0$ in a point $x_0 \in \mathbb{R}^n$. General theory of smooth differential equations guarantees that such solution exists and is unique for any x_0, at least for small $|t|$. Moreover, if $f \in C^k(\mathbb{R}^n, \mathbb{R}^n)$, then each $x_0 \in \mathbb{R}^n$ has a neighbourhood $U_0 \in \mathbb{R}^n$ such that for all $x \in U_0$ the solutions $\varphi^t(x)$ are defined on a common open interval around $t = 0$. Considered as functions of x, these solutions specify for each fixed t a map

$$\varphi^t : U_0 \to \mathbb{R}^n, \;\; x \mapsto \varphi^t(x), \tag{1.2}$$

that has continuous mixed partial derivatives with respect to all components of x up to (and including) order k.

The family of maps φ^t is called the (local) *flow* of (1.1). It has the following obvious properties:

$$\varphi^0(x) = x, \;\; \varphi^{t+s}(x) = \varphi^t(\varphi^s(x)),$$

where the second equality holds whenever both its sides are defined. Any family of maps that satisfies these conditions is called a (continuous-time) *dynamical system*.

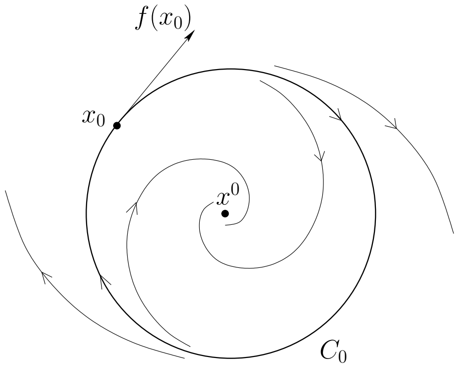


Figure 1.1: A sketch of a phase portrait with an equilibrium x^0 and a limit cycle C_0.

When t varies, the point $\varphi^t(x)$ traces a curve in the state space $\mathbb{R}^n$ that is called an *orbit* of (1.1). A *trivial orbit* can consist of one point. The vector $f(x)$ in the right-hand side of (1.1) is tangent to the nontrivial orbit at point x. Nontrivial orbits are oriented by the time advance and do not intersect. The collection of all orbits of an ODE system is called the *phase portrait*. Of course, while presenting phase portraits, one draws only (pieces of) few principal orbits (see, for example, Figure 1.1).

The important aim of the mathematical analysis of (1.1) is, thus, to describe the structure of the phase portrait. This cannot be achieved in general. However, a combination of analytical and numerical methods usually allows to establish the principal qualitative features of phase portraits of (1.1).

To describe a phase portrait, we first focus on *invariant sets* of (1.1), which are composed of orbits. The simplest invariant sets are *equilibria*, i.e. trivial orbits consisting of just one point: $x^0 = \varphi^t(x^0)$ for all t (see Figure 1.1). More interesting are *periodic orbits* or *cycles*, i.e. nontrivial orbits corresponding to periodic solutions

$$\varphi^{t+T_0}(x_0) = \varphi^t(x_0), \quad t \in \mathbb{R},$$

with (minimal) period $T_0 > 0$; these orbits are closed curves in $\mathbb{R}^n$. Isolated periodic orbits are called *limit cycles* (see Figure 1.1). Further classification of invariant sets distinguishes (smooth and non-smooth) *invariant manifolds* and *fractal invariant sets*.

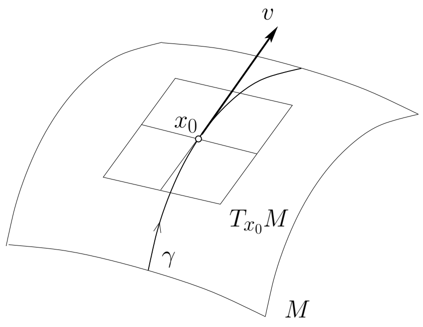


Figure 1.2: A smooth manifold (surface) $M \subset \mathbb{R}^3$ and its tangent space (plane) $T_{x_0}M$ at point x_0.

In general, a *manifold* $M \subset \mathbb{R}^n$ is a set of points satisfying

$$F(x) = 0,$$

where $F : \mathbb{R}^n \to \mathbb{R}^k$ is smooth and $k < n$. Consider a smooth curve $\mathit{\Gamma}$ in M parameterized as $x = x(t)$ and passing at $t = 0$ through $x(0) = x_0 \in M$. Its velocity vector $v(t) = \dot{x}(t)$ satisfies at $t = 0$ the equation

$$F_x(x_0)v = 0, \tag{1.3}$$

where F_x denotes the *Jacobian matrix* of F, i.e. the matrix with elements

$$[F_x]_{ij} = \frac{\partial F_i(x)}{\partial x_j}, \;\; i = 1, 2, \ldots, k; \; j = 1, 2, \ldots, n.$$

For a *smooth manifold*, the system (1.3) has $(n-k)$ linearly-independent solutions for each $x_0 \in M$. The collection of all solutions to (1.3) forms a linear subspace of $\mathbb{R}^n$ and is called the *tangent space* to M at x_0. Locally, a smooth manifold can be approximated by its tangent space and then parameterized by $\mathbb{R}^{n-k}$; $\dim M =$

$n - k$. The correspondence map between two local parameterizations of M near two close points is smooth. By definition, k is the *codimension* of M; $\text{codim}\, M = k$. Examples of two-dimensional (codimension-one) smooth manifolds in $\mathbb{R}^3$ are surfaces: e.g. cylinders, spheres and tori.

Systems (1.1) can have invariant tori, spheres, etc. Another important example of an invariant manifold is a set composed by all orbits approaching an equilibrium or a cycle as $t \to \pm\infty$ (*stable and unstable invariant manifolds*). A restriction of the dynamical system defined by (1.1) to its smooth invariant manifolds is a dynamical system on this manifold that is locally represented by a system of p smooth ODEs.

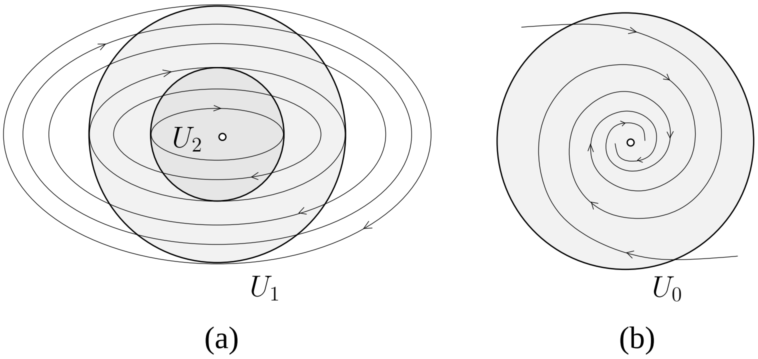


Figure 1.3: Stable (a) and asymptotically stable (b) equilibrium.

A bounded invariant set can be stable or unstable. A bounded invariant set is called *stable* if for any neighbourhood U_1 of the set there is another neighbourhood U_2 such that all solutions starting in U_2 will remain in U_1 for all $t > 0$. If, moreover, there is a neighbourhood U_0 such that all solutions starting in U_0 tend to the invariant set as $t \to +\infty$, the set is called *asymptotically stable* or *attracting*. A fractal asymptotically stable invariant set is called a *strange attractor*.

Therefore, the structure of a phase portrait depends on the number and stability of equilibria, cycles, and other invariant sets, as well as their relative position.

In most applications, however, the system of differential equations depends on parameters $\alpha_1, \alpha_2, \dots, \alpha_k$. If we collect these parameters in a vector

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} \in \mathbb{R}^m,$$

we can write a parameter-dependent version of (1.1):

$$\dot{x} = f(x, \alpha), \quad x \in \mathbb{R}^n,\ \alpha \in \mathbb{R}^m, \tag{1.4}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. We assume that f is smooth in all its arguments. Now do we not only want to classify solutions of this system but also to understand how they depend on the parameters. Obviously, in this case the flow φ_α^t of (1.4) and its phase portrait are parameter-dependent.

When parameters are allowed to vary, the structure of the phase portrait of (1.4) can change qualitatively at some parameter value α_0. This phenomenon is called a *bifurcation*. Appearance and disappearance of equilibria and cycles and their stability changes are bifurcations. Probably the most well-known of them is the *Andronov-Hopf bifurcation*, when an equilibrium loses stability by giving rise to a small stable limit cycle.

In general, the parameter space $\mathbb{R}^m$ of (1.4) can be subdivided into subsets with topologically equivalent phase portraits. The boundaries of such subsets correspond to bifurcations and in the simplest cases are smooth manifolds. For a generic system (1.4) with a sufficient number of parameters, the *codimension* of a bifurcation is the difference between the dimension of the bifurcation boundary and the dimension of the whole parameter space. In other words, the codimension of a bifurcation is the number of control parameters of a generic system (1.4) that one has to tune to meet this bifurcation. A collection of the bifurcation boundaries together with all non-equivalent phase portraits is called the *bifurcation diagram*. The bifurcation diagram, when constructed, provides very complete information about possible solutions of (1.4) and their rearrangements under parameter variations. We note that even partial knowledge of the bifurcation diagram gives important insights on system dynamics.

1.2 Equilibria and their bifurcations

In this section we give a brief survey on the theory of equilibria and equilibrium bifurcations in ODE systems with one and two control parameters. For more details we refer to [86, Chapters 3, 5 and 8].

To fix notation, introduce a system of autonomous ordinary differential equations

$$\dot{x} \equiv \frac{dx}{dt} = f(x, \alpha), \tag{1.5}$$

where $x \in \mathbb{R}^n$ is a state (phase) variable vector, $\alpha \in \mathbb{R}^m$ is a parameter vector and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is sufficiently smooth.

1.2.1 Equilibria and their stability

Given $\alpha_0 \in \mathbb{R}^m$, an *equilibrium of* (1.5) is a solution $x_0 \in \mathbb{R}^n$ to the equation

$$f(x, \alpha_0) = 0. \tag{1.6}$$

If the Jacobian matrix $A = f_x(x_0, \alpha_0)$ has no eigenvalue λ with $\Re(\lambda) = 0$ (where $\Re()$ means 'the real part of'), then the equilibrium is called *hyperbolic*. The phase portrait of (1.5) near a hyperbolic equilibrium x_0 does not differ qualitatively from the phase portrait of $\dot{y} = Ay$ near $y = 0$. In particular, x_0 is locally asymptotically stable if all eigenvalues of A satisfy $\Re(\lambda) < 0$ and it is unstable if at least for one eigenvalue holds $\Re(\lambda) > 0$.

1.2.2 Generic codim 1 bifurcations of equilibria

Any hyperbolic equilibrium depends smoothly on system parameters and can be continued with respect to a component of α. Indeed, assuming that (1.5) depends only on one control parameter $\alpha \in \mathbb{R}$, we see that the equilibria satisfy the equation

$$f(x, \alpha) = 0, \tag{1.7}$$

where $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with $f(x_0, \alpha_0) = 0$.

Varying α and tracing the curve defined by (1.7) in the (x, α)-space, we may encounter a critical parameter value where the equilibrium loses hyperbolicity, so that either

(i) f_x has an eigenvalue $\lambda_1 = 0$; or
(ii) f_x has a pair of purely imaginary eigenvalues $\lambda_{1,2} = \pm i\omega_0$ with $\omega_0 > 0$.

Both cases lead to *equilibrium bifurcations* when the parameter α crosses its critical value: the local phase portrait changes qualitatively. Actually, essential changes occur on a parameter-dependent smooth invariant curve in case (i) and on a parameter-dependent smooth invariant surface in case (ii), so-called *center manifolds*. Nearby orbits approach these manifolds exponentially fast as time grows or decays. The number of stable (unstable), directions is determined by the number of eigenvalues of A with $\Re(\lambda) < 0$ ($\Re(\lambda) > 0$).

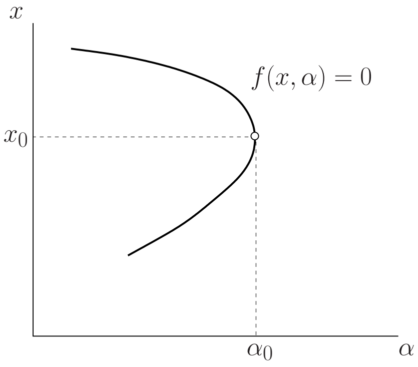


Figure 1.4: An equilibrium curve with an LP.

Limit point (fold) bifurcation

Assume that case (i) occurs at α_0 and that x_0 is the corresponding critical equilibrium. Generically, the equilibrium curve (1.7) turns quadratically at (x_0, α_0) (see Figure 1.4). The system (1.5) exhibits a *limit point* (LP) (or *fold*) *bifurcation*: two equilibria collide and disappear at the critical parameter value.

In generic one-parameter ODEs, the eigenvalue $\lambda_1 = 0$ is algebraically simple at the LP-bifurcation and no other eigenvalues with $\Re(\lambda) = 0$ are present. At α_0, there is a smooth one-dimensional invariant manifold (curve) W_0^c in the phase space of (1.5) called the *critical center manifold* at the LP-bifurcation. This curve can be locally parameterized by $u \in \mathbb{R}$ with small $|u|$. The restriction of (1.5) to W_0^c has the form

$$\dot{u} = au^2 + O(|u|^3), \tag{1.8}$$

where a is the *quadratic normal form coefficient* for the LP bifurcation.

Andronov-Hopf bifurcation

When case (ii) occurs at α_0, the critical equilibrium x_0 has a pair of purely imaginary eigenvalues $\lambda_{1,2} = \pm i\omega_0,\ \omega_0 > 0$. Generically, an *Andronov-Hopf bifurcation* (H) occurs: a branch of periodic solutions to (1.5) with $O(\sqrt{|\alpha - \alpha_0|})$-amplitude is born, either for $\alpha < \alpha_0$ or $\alpha > \alpha_0$.

Considering only generic one-parameter systems (1.5), we can assume that the eigenvalues $\lambda_{1,2}$ are simple and are the only critical eigenvalues of $A = f_x(x_0, \alpha_0)$ with $\Re(\lambda) = 0$. Due to the Implicit Function Theorem, a unique smooth continuation $x_0(\alpha)$ of x_0 exists. However, generically, its stability type changes, since a pair of complex-conjugate eigenvalues of $A(\alpha) = f_x(x_0(\alpha), \alpha)$,

$$\lambda_{1,2}(\alpha) = \beta(\alpha) \pm i\omega(\alpha)$$

passes through the imaginary axis: $\beta(\alpha_0) = 0, \beta'(\alpha_0) \neq 0$, and $\omega(\alpha_0) = \omega_0$.

In this case, at α_0, there is a smooth two-dimensional invariant manifold (surface) W_0^c in the phase space of (1.5) that can be locally parameterized by $z \in \mathbb{C}$ with small $|z|$. This is the *critical center manifold* at the Hopf bifurcation. The restriction of (1.5) to W_0^c can be transformed by smooth invertible coordinate transformations to the form

$$\dot{z} = i\omega_0 z + c_1 z^2 \bar{z} + O(|z|^4), \tag{1.9}$$

where $l_1 = \Re(c_1)$ is called the *first Lyapunov coefficient* at the Hopf bifurcation. The equilibrium x_0 is (nonlinearly) stable within the manifold W_0^c if $l_1 < 0$ and is unstable if $l_1 > 0$. This becomes obvious if we write (1.9) in the polar coordinates (ρ, φ), so that $z = \rho e^{i\varphi}$:

$$\begin{cases} \dot{\rho} = l_1\rho^3 + O(\rho^4), \\ \dot{\varphi} = \omega_0 + O(\rho^2). \end{cases} \tag{1.10}$$

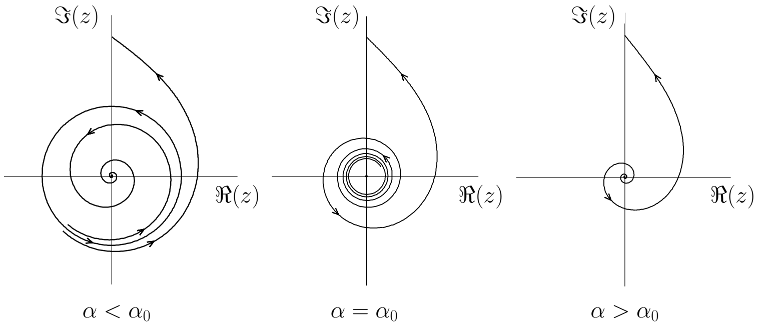


Figure 1.5: Subcritical Hopf bifurcation with $l_1(\alpha_0) < 0$ and $\beta'(\alpha_0) > 0$.

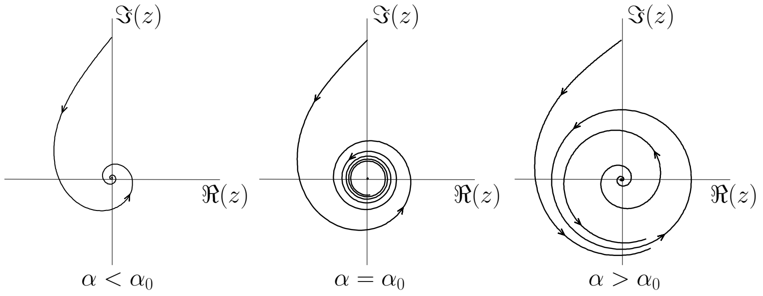


Figure 1.6: Supercritical Hopf bifurcation with $l_1(\alpha_0) < 0$ and $\beta'(\alpha_0) > 0$.

The periodic solution is born unstable for $\alpha < \alpha_0$ if $l_1(\alpha_0) > 0$, in which case the bifurcation is called subcritical. The appearing periodic solution is born stable for $\alpha > \alpha_0$ if $l_1(\alpha_0) < 0$; then the bifurcation is called supercritical. (see Figures 1.5 and 1.6, respectively).

1.2.3 Generic codim 2 bifurcations of equilibria

In generic systems (1.5) depending on two control parameters, the bifurcation boundaries corresponding to fold and Hopf bifurcations are curves in the parameter plane. These curves can have special points, in a neighbourhood of which the bifurcation diagram exhibits new features. These points correspond to codim 2 bifurcations of equilibria. At such a point α_0, the system (1.5) has an equilibrium x_0, for which one of the following five cases holds:

CP: $\lambda_1 = 0,\ a = 0$ (*cusp*);
GH: $\lambda_{1,2} = \pm i\omega_0,\ l_1 = 0$ (*Bautin*);
BT: $\lambda_1 = \lambda_2 = 0$ (*Bogdanov-Takens*);
ZH: $\lambda_1 = 0,\ \lambda_{2,3} = \pm i\omega_0, \omega_0 > 0$ (*fold-Hopf*);
HH: $\lambda_{1,2} = \pm i\omega_1,\ \lambda_{3,4} = \pm i\omega_2, \omega_{1,2} > 0$ (*double Hopf*).

Here λ_k are the critical eigenvalues of $A = f_x(x_0, \alpha_0)$ which are assumed to be simple and the only eigenvalues with $\Re(\lambda) = 0$; their number determines the dimension of the parameter-dependent invariant center manifold W^c_α near the bifurcation. Finally, the coefficients a and l_1 are those appearing in the critical normal forms (1.8) and (1.10), respectively.

1.2.4 Branch points

A *branch point* is a point where the Jacobian matrix $f_{x,\alpha}(x, \alpha)$ of (1.7) is rank deficient. This is a non-generic situation in one-parameter problems where the Implicit Function Theorem cannot be applied to ensure the existence of a unique smooth branch of solutions. And although it is encountered often in practical problems that exhibit some form of symmetry (equivariance), we will not go deeper into this subject in this thesis.

1.3 Limit Cycles and their bifurcations

In this section we give a brief survey on the theory of periodic solutions and their bifurcations. Consider again a system of ODEs:

$$\dot{x} \equiv \frac{dx}{dt} = f(x, \alpha) \tag{1.11}$$

where $x \in \mathbb{R}^n$ is a phase variable vector, $\alpha \in \mathbb{R}^m$ is a parameter vector and f is a sufficiently smooth function.

1.3.1 Cycles and their stability

Given $\alpha_0 \in \mathbb{R}^m$, a *cycle of* (1.11) is a closed orbit C_0 corresponding to a *periodic solution* $x_0(t) = x_0(t + T_0)$, where $T_0 > 0$ is its (minimal) period. Such solution satisfies the following boundary value problem (BVP):

$$\left\{ \begin{array}{rcl} \dot{x}(t) - f(x(t), \alpha_0) & = & 0, \ t \in [0, T_0], \\ x(0) - x(T_0) & = & 0, \\ \int_0^{T_0} \langle x(\tau), \dot{y}(\tau) \rangle \, d\tau & = & 0, \end{array} \right. \tag{1.12}$$

where y is a *reference periodic solution*. The last equation in (1.12) selects one periodic solution from infinitely many periodic solutions satisfying the first two equations but having different initial points on C_0. This equation, called the *integral phase condition*, selects a solution that has the minimal L_2-distance (2-norm distance) to y among all phase-shifted periodic solutions. Notice that the period T_0 is not given and should be found from (1.12) together with the periodic solution $x_0(t)$.

Introduce $A(t) = f_x(x_0(t), \alpha)$ and consider the initial-value problem for the fundamental matrix solution $Y(t)$, namely

$$\frac{dY}{dt} = A(t)Y, \ \ Y(0) = I_n, \tag{1.13}$$

where I_n is the $n \times n$ identity matrix. The *monodromy matrix* $M = Y(T_0)$ always has a “trivial” eigenvalue $\mu = 1$. The cycle C_0 is called *hyperbolic* if there are no other eigenvalues with $|\mu| = 1$, and non-hyperbolic otherwise. Note that a hyperbolic cycle is isolated in the phase space, meaning that there is no other periodic orbit near C_0. Often, the eigenvalues of M are called *multipliers of* C_0.

The phase portrait of (1.11) near a hyperbolic cycle C_0 does not differ qualitatively from the phase portrait of $\dot{y} = A(t)y$ near $y = 0$ in the space $(y, t \bmod T_0)$. In particular, the cycle is locally asymptotically stable if all nontrivial eigenvalues of M satisfy $|\mu| < 1$ and it is unstable if at least for one eigenvalue holds $|\mu| > 1$.

1.3.2 Generic codim 1 bifurcations of cycles

Any hyperbolic cycle depends smoothly on system parameters and can be continued with respect to a component of α. Assuming that (1.11) depends only on one parameter $\alpha \in \mathbb{R}$, we can consider the BVP

$$\left\{ \begin{array}{rcl} \dot{x}(t) - f(x(t), \alpha) & = & 0, \ t \in [0, T], \\ x(0) - x(T) & = & 0, \\ \int_0^T \langle x(\tau), \dot{y}(\tau) \rangle \, d\tau & = & 0, \end{array} \right. \tag{1.14}$$

which has a T_0-periodic solution $x_0(t)$ at α_0. If $\mu = 1$ is simple, this BVP defines a branch of $T(\alpha)$-periodic solutions $x_\alpha(t)$, and thus a continuation of the cycle C_0 with respect to α. In the continuation context, the reference periodic solution $y(t)$ is usually a periodic solution found at some previous value of the parameter.

Varying α and tracing the cycle, we may encounter a critical parameter value, where the cycle loses hyperbolicity, so that either

(i) M has a multiple eigenvalue $\mu_1 = 1$; or
(ii) M has an eigenvalue $\mu_1 = -1$; or
(iii) M has a pair of purely imaginary eigenvalues $\mu_{1,2} = \mathrm{e}^{\pm i\theta_0}$ with $0 < \theta_0 < \pi$.

All three cases lead to *cycle bifurcations*, when the parameter α crosses its critical value: the local phase portrait near the critical cycle changes qualitatively.

Limit point (fold) of cycles

Assume that case (i) occurs at α_0 and that C_0 is the corresponding critical cycle with its T_0-periodic solution $x_0 = x_0(t)$. Generically, the cycle manifold in the

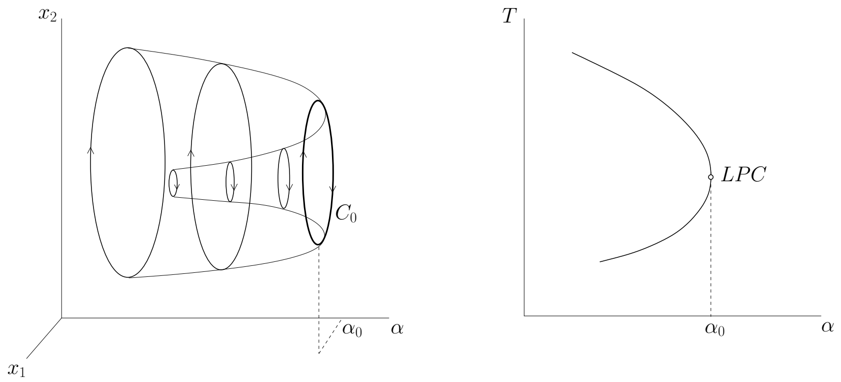


Figure 1.7: Cycle manifold with an LPC-point.

(x, T, α)-space turns quadratically at α_0 (see Figure 1.7). The system (1.5) exhibits a *limit point cycle* (LPC) (or *periodic fold*) *bifurcation*: two cycles collide and disappear at the critical parameter value.

Period-doubling (flip) bifurcation

Assume that case (ii) occurs at α_0 and that C_0 is the corresponding critical cycle with its T_0-periodic solution $x_0 = x_0(t)$. Generically, at $\alpha = \alpha_0$, this cycle changes stability and another cycle C_1 with minimal period $T_1(\alpha)$ bifurcates from it for $\alpha > \alpha_0$ or $\alpha < \alpha_0$ (see Figure 1.8). Since $T_1(\alpha) \to 2T_0$ as $\alpha \to \alpha_0$, the bifurcation is called the *period-doubling bifurcation* (PD) or *flip bifurcation*: a cycle of the approximately double period appears at the critical parameter value.

Neimark-Sacker (torus) bifurcation

Finally, assume that case (iii) occurs at α_0 and that C_0 is the corresponding critical cycle with its T_0-periodic solution $x_0 = x_0(t)$. Generically, at $\alpha = \alpha_0$, the cycle changes stability and an *invariant two-dimensional torus* bifurcates from it for $\alpha > \alpha_0$ or $\alpha < \alpha_0$ (see Figure 1.9). Its “thickness” is $\sim \sqrt{|\alpha - \alpha_0|}$. This

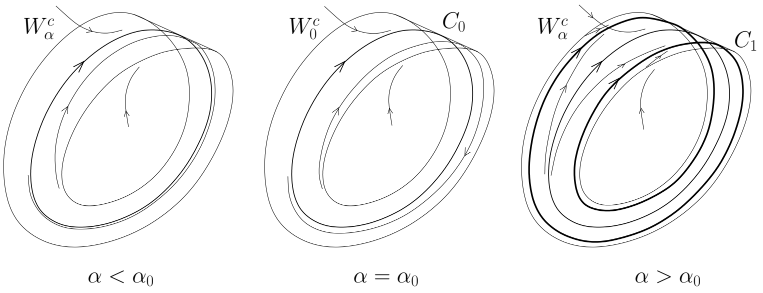


Figure 1.8: PD-bifurcation in $\mathbb{R}^3$.

torus has finite smoothness increasing as $\alpha \to \alpha_0$. Orbits on this torus can be periodic, non-periodic but tending to periodic ones, or quasi-periodic. Generically, when α passes α_0, an infinite number of LPC-bifurcations of cycles with large periods occur. These bifurcations delimit open intervals of parameter α, where two hyperbolic cycles coexist on the torus. Outside these intervals, quasiperiodic orbits fill the torus. This bifurcation is called the *Neimark-Sacker bifurcation* (NS) or *torus bifurcation*: an invariant torus around the primary cycle appears at the critical parameter value, while the cycle itself changes its stability type.

1.3.3 Generic codim 2 bifurcations of cycles

In generic systems (1.5) depending on two control parameters the bifurcation boundaries corresponding to fold, period-doubling, and Neimark-Sacker bifurcations of limit cycles are curves in the parameter plane. Codimension 2 bifurcation points can be met along these curves. At such a point α_0, the system (1.5) has a cycle C_0, for which one of the following eleven cases holds:

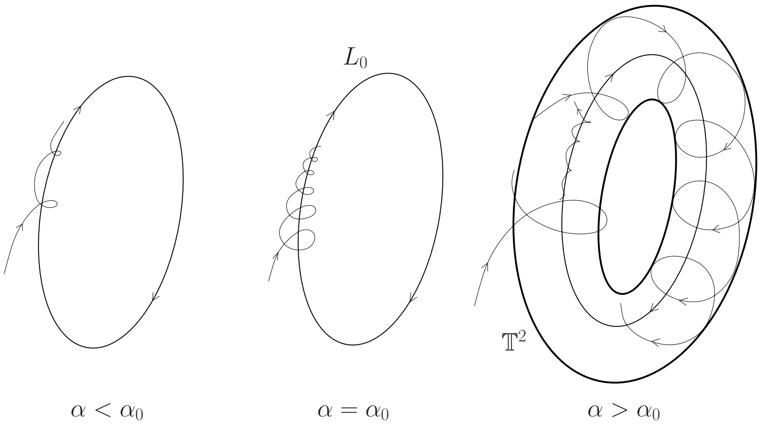


Figure 1.9: NS-bifurcation in $\mathbb{R}^3$.

CPC	:	$\mu_1 = 1,\ a = 0$ (*cusp of cycles*);
GPD	:	$\mu_1 = 1,\ b = 0$ (*generalized period-doubling*);
CH	:	$\mu_{1,2} = \mathrm{e}^{\pm i\theta_0},\ \Re(d) = 0$ (Chenciner bifurcation);
R1	:	$\mu_1 = \mu_2 = 1$ (1:1 *resonance*);
R2	:	$\mu_1 = \mu_2 = -1$ (1:2 *resonance*);
R3	:	$\mu_{1,2} = \mathrm{e}^{\pm i\theta_0},\ \theta_0 = \frac{2\pi}{3}$ (1:3 *resonance*);
R4	:	$\mu_{1,2} = \mathrm{e}^{\pm i\theta_0},\ \theta_0 = \frac{\pi}{2}$ (1:4 *resonance*);
LPPD	:	$\mu_1 = 1,\ \mu_2 = -1$ (*fold-flip*);
LPNS	:	$\mu_1 = 1,\ \mu_{2,3} = \mathrm{e}^{\pm i\theta_0}$ (*fold–Neimark-Sacker*);
PDNS	:	$\mu_1 = -1,\ \mu_{2,3} = \mathrm{e}^{\pm i\theta_0}$ (*flip–Neimark-Sacker*);
NSNS	:	$\mu_{1,2} = \mathrm{e}^{\pm i\theta_1},\ \mu_{3,4} = \mathrm{e}^{\pm i\theta_2}$ (*double Neimark-Sacker*).

Here μ_k are the non-trivial critical eigenvalues of the monodromy matrix M of C_0, which are assumed to be simple and the only eigenvalues with $|\mu| = 1$. It is also assumed that $0 < \theta_k < \pi$ for $k = 0, 1, 2$. The coefficients a, b, and d appear in the critical periodic normal forms of the LPC, PD and NS bifurcations (which are not listed here, since they are of less importance for the rest of this thesis).

1.3.4 Branching of cycles

A branch of periodic solution $(x(\cdot), T, \alpha)$ defined by (1.14) can intersect another branch of periodic solutions at some critical parameter value α_0, so that both solutions coincide. This never happens in generic one-parameter ODEs and although it is often encountered if (1.11) has some symmetry, we do not go deeper into this issue in this thesis.

1.4 Homoclinic orbits

In this section we give a brief survey on the theory of bifurcations of orbits homoclinic to equilibria in generic ODEs depending on one parameter. We also list some codim 2 homoclinic bifurcations appearing in generic ODE systems with two control parameters. For more details we refer to [86, Chapter 6]. For references, let us reintroduce

$$\dot{x} = f(x, \alpha), \tag{1.15}$$

where $x \in \mathbb{R}^n, \alpha \in \mathbb{R}^k$ and f is sufficiently smooth.

1.4.1 (Un)stable manifolds and homoclinic orbits

Fix $\alpha = \alpha_0$ and let $x_0 \in \mathbb{R}^n$ be an equilibrium of (1.15). The invariant sets

$$\begin{array}{rcl} W^s(x_0) & = & \{x \in \mathbb{R}^n : \varphi^t_{\alpha_0}(x) \to x_0 \text{ if } t \to +\infty\}, \\ W^u(x_0) & = & \{x \in \mathbb{R}^n : \varphi^t_{\alpha_0}(x) \to x_0 \text{ if } t \to -\infty\} \end{array}$$

are called the *stable* and *unstable sets of* x_0, respectively. The set $W^s(x_0)$ consists of all orbits approaching x_0 forward in time, while $W^u(x_0)$ consists of all orbits approaching x_0 backward in time.

If x_0 is hyperbolic then both sets are smooth manifolds (hypersurfaces) in $\mathbb{R}^n$. If $A = f_x(x_0, \alpha_0)$ has n_s eigenvalues μ with $\Re(\mu) < 0$ and n_u eigenvalues λ with $\Re(\lambda) > 0$, then $\dim W^s(x_0) = n_s$ and $\dim W^u(x_0) = n_u$ with $n_s + n_u = n$. Moreover, these manifolds are tangent at x_0 to the (generalized) eigenspaces T^s and T^u, corresponding to all eigenvalues μ with $\Re(\mu) < 0$ and λ with $\Re(\lambda) > 0$, respectively.

If x_0 is non-hyperbolic, the stable and unstable sets become smooth submanifolds (hypersurfaces with borders) already near x_0. If A has n_s eigenvalues μ with $\Re(\mu) < 0$, n_c eigenvalues λ^c with $\Re(\lambda^c) = 0$, and n_u eigenvalues λ with $\Re(\lambda) > 0$, so that $n_s + n_c + n_u = n$, then $\dim W^s(x_0) \leq n_s + n_c$ and $dimW^u(x_0) \leq n_u + n_c$. The exact dimensions of these submanifolds are determined by the orbit behaviour on the n_c-dimensional center manifold $W^c(x_0)$.

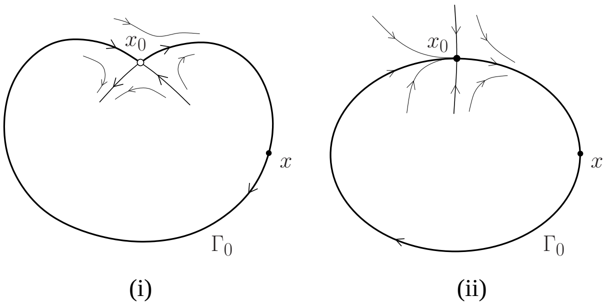


Figure 1.10: Homoclinic orbits to equilibria in planar systems: (i) x_0 is a saddle with eigenvalues $\mu_1 < 0 < \lambda_1$; (ii) x_0 is a saddle-node with eigenvalues $\lambda^c = 0, \mu_1 < 0$.

An orbit $\mathit{\Gamma}_0 \neq x_0$ is called *homoclinic* to x_0 if $\mathit{\Gamma}_0 \subset W^s(x_0) \cap W^u(x_0)$, i.e. $\varphi^t_{\alpha_0}(x) \to x_0$ as $t \to \pm\infty$ for any $x \in \mathit{\Gamma}_0$. Figure 1.10 illustrates two types of homoclinic orbits in planar systems ($n = 2$).

Given an equilibrium x_0, a homoclinic solution $x^0(t)$ satisfies the following *homoclinic problem*:

$$\left\{\begin{array}{rcl} \dot{x}(t) - f(x(t), \alpha_0) & = & 0, \quad t \in \mathbb{R}, \\ x(t) & \to & x_0, \quad t \to \pm\infty, \\ \displaystyle\int_{-\infty}^{+\infty} \langle x(\tau) - y(\tau), \dot{y}(\tau) \rangle \, d\tau & = & 0, \end{array}\right. \tag{1.16}$$

where y is a reference solution that is assumed to be given and whose derivative decays sufficiently fast as $t \to \pm\infty$.The integral equation in (1.16) is a condition for a local minimum of the L_2-distance (2-norm distance) between x and y over time shifts.

1.4.2 Homoclinic orbits to hyperbolic equilibria

Let x_0 be a hyperbolic equilibrium of (1.15) at α_0. Denote by μ_j, for $j = 1, 2, \dots, n_s$, the *stable eigenvalues* of $A = f_x(x_0, \alpha_0)$ satisfying $\Re(\mu) < 0$ and by λ_j, for $j = 1, 2, \dots, n_u$, the *unstable eigenvalues* of $A = f_x(x_0, \alpha_0)$ satisfying $\Re(\lambda) > 0$. Stable eigenvalues with maximal $\Re(\mu)$ are called *leading stable eigenvalues*, while unstable eigenvalues with minimal $\Re(\lambda)$ are called *leading unstable eigenvalues*. Among all (stable and unstable) leading eigenvalues, the ones with minimal $|\Re(\lambda)|$ or $|\Re(\mu)|$ are called *determining*.

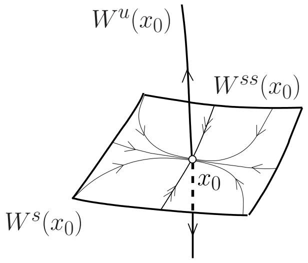


Figure 1.11: Unstable, stable, and strong stable invariant manifolds of a saddle equilibrium in $\mathbb{R}^3$.

Almost all orbits on $W^s(x_0)$ tend to x_0 as $t \to +\infty$ along the generalized eigenspace of A corresponding to the stable leading eigenvalues (μ_1). Exceptional orbits form a *nonleading* or *strong stable manifold* $W^{ss}(x_0)$ (see Figure 1.11). Similarly, all orbits on $W^u(x_0)$, except those belonging to a *nonleading* or *strong unstable manifold* $W^{uu}(x_0)$, tend to x_0 as $t \to -\infty$ along the generalized eigenspace of A corresponding to the unstable leading eigenvalues (λ_1).

If there is a single simple stable/unstable eigenvalue, then we can introduce four singular vectors, $q_1^{s,u}$ and $p_1^{s,u}$, corresponding to the leading eigenvalues μ_1

and λ_1 of the matrices A and A^{T}:

$$Aq_1^s = \mu_1 q_1^s, \;\; Aq_1^u = \lambda_1 q_1^u$$

and

$$A^{\mathrm{T}} p_1^s = \mu_1 p_1^s, \;\; A^{\mathrm{T}} p_1^u = \lambda_1 p_1^u.$$

Using these vectors, the requirement for the homoclinic orbit $\mathit{\Gamma}_0$ to depart and return along the leading directions can be formulated as

$$\lim_{t\to+\infty} \mathrm{e}^{-\mu_1 t}\langle p_1^s, x^0(t) - x_0\rangle \neq 0 \tag{1.17}$$

and

$$\lim_{t\to-\infty} \mathrm{e}^{-\lambda_1 t}\langle p_1^u, x^0(t) - x_0\rangle \neq 0. \tag{1.18}$$

Another condition for the hyperbolic homoclinic bifurcation is related to the global twistedness of the manifolds $W^s(x_0)$ and $W^u(x_0)$ around the homoclinic orbit $\mathit{\Gamma}_0$.

Define $A(t) = f_x(x^0(t), \alpha_0)$ and introduce the following problem

$$\left\{ \begin{array}{rcl} \dot{\varphi}(t) + A^{\mathrm{T}}(t)\varphi(t) & = & 0, \;\; t \in \mathbb{R}, \\ \varphi(t) & \to & 0, \;\; t \to \pm\infty, \\ \int_{-\infty}^{+\infty} \langle \varphi(\tau) - \psi(\tau), \dot{\psi}(\tau)\rangle \, d\tau & = & 0, \end{array} \right. \tag{1.19}$$

where ψ is a reference function. The third equation in (1.19) selects one solution out of the family $c\varphi(t)$ for $c \in \mathbb{R}$. The solution φ exists and is unique. The limit behaviour of $\varphi(t)$ as $t \to \pm\infty$ determines the twistedness of the space $Z(t)$ around the homoclinic orbit $\mathit{\Gamma}_0$, where $Z(t)$ is the sum of the linear spaces tangent to $W^s(x_0)$ and $W^u(x_0)$. Generically, both

$$\lim_{t\to-\infty} \mathrm{e}^{\mu_1 t}\langle q_1^s, \varphi(t)\rangle \neq 0 \tag{1.20}$$

and

$$\lim_{t\to+\infty} \mathrm{e}^{-\lambda_1 t}\langle q_1^u, \varphi(t)\rangle \neq 0. \tag{1.21}$$

The non-degeneracy condition (1.20) is illustrated for $n = 3$ in Figure 1.12, where the saddle x_0 has one unstable and two stable eigenvalues. In this case,

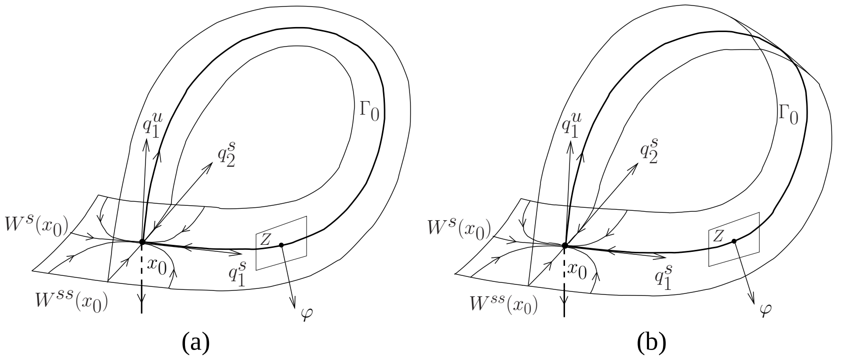


Figure 1.12: Simple (a) and twisted (b) stable manifold at a saddle homoclinic bifurcation in $\mathbb{R}^3$.

$W^s(x_0)$ is two-dimensional and its continuation backward in time intersects itself along the non-leading stable manifold $W^{ss}(x_0)$. Therefore, generically, that part of the stable manifold $W^s(x_0)$, to which Γ_0 belongs, is either a simple or a twisted (Möbius) non-smooth band.

Possible types of homoclinic orbits to hyperbolic equilibria are:

- The saddle homoclinic orbit: the leading stable (μ_1) and unstable (λ_1) eigenvalues are real.
- The saddle-focus homoclinic orbit: the leading unstable eigenvalue is real, while the leading stable eigenvalues are complex-conjugate, or vice versa.
- The focus-focus homoclinic orbit: both stable and unstable leading eigenvalues are complex.

1.4.3 Homoclinic orbits to saddle-node equilibria

Only homoclinic orbits to nonhyperbolic equilibria with one eigenvalue $\lambda^c = 0$ appear in generic one-parameter systems (1.15): to create a homoclinic orbit to a nonhyperbolic equilibrium with a pair of purely-imaginary eigenvalues, one needs to tune at least two parameters. On the contrary, an appearance of a homoclinic orbit $\mathit{\Gamma}_0$ to an equilibrium x_0 that has a single critical eigenvalue $\lambda^c = 0$ is not an extra degeneracy, since the stable set $W^s(x_0)$ can intersect with the unstable set $W^u(x_0)$ transversally. Actually, the appearance of a finite number of homoclinic orbits $\mathit{\Gamma}_0, \mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_{N-1}$ does not increase the degeneracy either. We assume such transversality further.

When $\lambda^c = 0$ is a critical eigenvalue of x_0, generically, a LP (fold) bifurcation happens when α crosses its critical value α_0 as described in Section 1.2.2. We can assume, therefore, that the genericity conditions for the LP-bifurcation hold. They imply that there exists a one-dimensional center manifold W^c_α that is exponentially attracting and/or repelling in transverse directions. When α passes α_0, two equilibria in W^c_α collide and disappear.

Finally, in generic systems (1.15), the homoclinic orbits $\mathit{\Gamma}_k$ depart and return to x_0 at the critical parameter value along the null-vector $q \in \mathbb{R}^n$ of $A = f_x(x_0, \alpha_0)$: $Aq = 0$. Such homoclinic orbits are called *central*. Introduce the adjoint null-vector $p \in \mathbb{R}^n : A^{\mathrm{T}}p = 0,\ \langle p, q\rangle = 1$. Then the non-degeneracy conditions that guarantee that $\mathit{\Gamma}_0$ is central are

$$\lim_{t\to+\infty} t^{-1}\langle p, x^0(t) - x_0\rangle \neq 0 \tag{1.22}$$

and

$$\lim_{t\to-\infty} t^{-1}\langle p, x^0(t) - x_0\rangle \neq 0. \tag{1.23}$$

If either of these conditions is violated, the homoclinic orbit is non-smooth.

Generically, there are still two cases: $N = 1$ and $N \geq 2$ (with N the number of homoclinics), but we will not go into more detail on this subject here.

1.4.4 Codim 2 bifurcations of homoclinic orbits

While following a curve of homoclinic orbits in system (1.15) with two control parameters, one can encounter several codim 2 bifurcations. In the simplest cases, such bifurcations are detectable by looking at the eigenvalues of the equilibrium x_0. In more complicated cases, asymptotic properties of the homoclinic solution or nearby solutions have to be taken into account.

Codim 2 homoclinic orbits to hyperbolic equilibria

The following codim 2 cases can be expected in generic two-parameter ODE systems along curves of homoclinic orbits to hyperbolic equilibria.

(1) *Neutral saddle or saddle-focus*. The saddle quantity vanishes

$$\sigma_0 = \Re(\mu_1) + \Re(\lambda_1) = 0.$$

(2) *Double real determining eigenvalue*. In this case, either μ_1 or λ_1 is double.

(3) *Three determining eigenvalues*: there is a simple real determining eigenvalue $_1$ and a simple pair of complex eigenvalues $_{2,3}$ such that

$$\Re(\lambda_1) = \Re(\lambda_{2,3}),$$

or

$$\Re(\mu_1) = \Re(\mu_{2,3}).$$

(4) *Non-hyperbolic equilibrium*:

$$\Re(\mu_1) = 0 \;\; \texttt{or} \;\; \Re(\lambda_1) = 0\,,$$

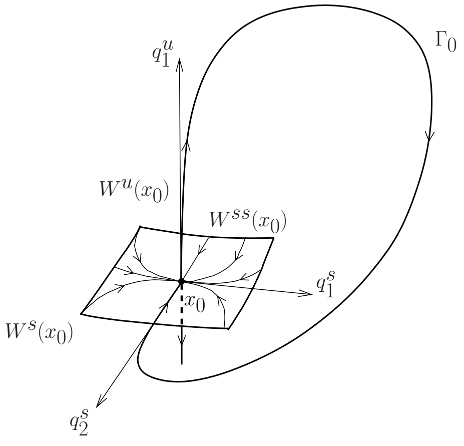


Figure 1.13: Orbit flip in $\mathbb{R}^3$.

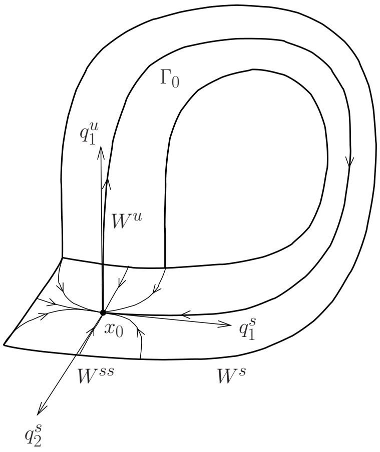


Figure 1.14: Inclination flip in $\mathbb{R}^3$.

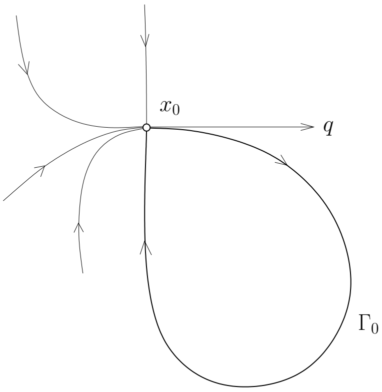


Figure 1.15: Non-central homoclinic orbit to a non-hyperbolic equilibrium in $\mathbb{R}^2$.

so that the equilibrium undergoes either a fold or a Hopf bifurcation. In the first case, we have a *non-central homoclinic-to-saddle-node* bifurcation (see below), while the second case is usually referred to as a *Shilnikov-Hopf* bifurcation.

(5) *Orbit flip*: either condition (1.17) or condition (1.18) does not hold for a saddle homoclinic orbit. If we have equality in (1.17), then the homoclinic orbit Γ_0 returns to the saddle x_0 along its non-leading stable manifold. If (1.18) does not hold, the homoclinic orbit Γ_0 departs from the saddle x_0 along its non-leading unstable manifold (see Figure 1.13).

(6) *Inclination flip*: the condition (1.20) or (1.21) is violated for a saddle homoclinic orbit. In this case, the homoclinic orbit Γ_0 has degenerate twistedness (see Figure 1.14 for an illustration).

(7) *Higher non-transversality*: it could happen that the intersection of the manifolds $W^s(x_0)$ and $W^u(x_0)$ along Γ_0 is not transversal with $\mathrm{codim} Z(t) = 2$.

This list is incomplete. Other cases, such as neutral focus-focus or neutrally-divergent saddle-focus, are possible but involve bifurcations of systems always having an infinite number of cycles. There are many other possibilities involving several homoclinic to x_0 orbits or orbits connecting two distinct equilibria.

Codim 2 homoclinic orbits to non-hyperbolic equilibria

The following codim 2 cases can be expected in generic two-parameter ODE systems along curves of homoclinic orbits to non-hyperbolic equilibria.

(1) *Non-central homoclinic orbit*: the condition (1.22) or (1.23) does not hold. In such a case, the homoclinic orbit to the equilibrium with zero eigenvalue is non-smooth (see Figure 1.15).

(2) *Non-transversal homoclinic orbit*: the stable and unstable sets $W^{s,u}(x_0)$ of the non-hyperbolic equilibrium x_0 have a quadratic tangency along the homoclinic orbit Γ_0.

Chapter 2

Neural models as dynamical systems

Dynamical systems are in use in numerous fields of science. One field where dynamical systems play an important role, is neurobiology, more precisely in the modelling of neurons. In this chapter, first the basics of neural modelling are explained, followed by a presentation of four of the most important and well-known neural models. Then a new classification of neural models is suggested, and finally the phase response curve is introduced.

2.1 Introduction to neural modelling

Most of this section is adapted from [47], Chapter 2.

2.1.1 Molecules, atoms, electrons and ions

It is now accepted as a scientific fact that all substances that we see around us consist of molecules. The number of different molecules is huge but they are all formed by combinations of atoms and the number of different atoms is relatively small - at this time 117. In chemistry these are called the elements and they are collected in the Periodic Table named after the Russian chemist Mendelejev

(1869).

An atom consists of a nucleus and a number of electrons in shells around that nucleus. The nucleus contains neutrons, which are electrically neutral, and protons, which have a basic electrical charge, identical for all protons and by convention said to be positive. This is called the elementary charge e, cf. Table 2.1. The electrons have the same amount of electrical charge but of an opposite nature and therefore called negative.

An atom is electrically neutral if the number of electrons in its shells is equal to the number of protons in its nucleus. However, many atoms tend either to lose one or more electrons or to acquire surplus electrons. If this happens the atom is called an ion. Some ions are very important in electrophysiology, namely:

- Sodium (Na from its Latin name *Natrium*) has atomic number 11, atomic weight 22.99 and valence 1. The latter means that it tends to lose one electron and so form the ion Na^+.

- Potassium (K from its Latin name *Kalium*) has atomic number 19, atomic weight 39.102 and valence 1. The latter means that it tends to lose one electron and so form the ion K^+.

- Calcium (Ca) has atomic number 20, atomic weight 40.08 and valence 2. The latter means that it tends to lose two electrons and so form the ion Ca^{2+}.

- Chlorine (Cl) has atomic number 17, atomic weight 35.45 and valence -1. The latter means that it tends to acquire a surplus electron and so form the ion Cl^-.

We note that the atomic number of an element is the number of protons in the nucleus; therefore it is always a positive integer number. The atomic weight is important in computations that involve concentrations. In the experimental literature the word ”concentration” has a precise technical meaning, namely the number of mol of a substance (the solute) that is solved in a kilogram of solvent. Its unit is the molar, i.e. one mol per kilogram, cf. Table 2.2. For example, a sodium ion solution with a concentration of one molar in one kilogram of solvent must contain 22.99 grams of sodium since 22.99 is the atomic weight of sodium. The number of

Definition	Abbreviation	Value
Avogadro's number	N	$6.02214 \times 10^{23}/mol$
Faraday's constant	F	$96485C/mol$
elementary charge	e	$1.60218 \times 10^{-19}C$
gas constant	R	$8.315J/(mol.K)$

Table 2.1: Some important constants. We recall that a mol of a substance is a quantity that contains as many elementary units (atoms or molecules) as there are atoms in 0.012 kg of carbon-12.

Quantity	Symbol	Unit	Symbol
Current	I	Ampère	A (=C/s)
Charge	Q	Coulomb	C
Capacitance	C	Farad	F (=C/V)
Potential	V	Volt	V (=J/C)
Voltage	U	Volt	V (=J/C)
Resistance	R	Ohm	Ω (=V/A)
Conductance	g	Siemens	S=mho (=1/Ω)
Concentration	[]	molar	M
Energy		Joule	J (=$\frac{kg.m^2}{s^2}$)

Table 2.2: Important electrical quantities and their units.

ions is then given by Avogadro's number N, cf. Table 2.1 and their total electrical charge is Ne=96485 Coulomb (cf. Faraday's constant in Table 2.1).

If a molecule consists of several atoms, it has a molecular weight which is obtained by simply adding the atomic weights of the constituting atoms. For example, water ($\mathrm{H_2O}$) contains two hydrogen atoms (H) with atomic weight 1.00797 and one oxygen atom (O) with atomic weight 15.9994. Therefore the molecular weight of water is $2 \times 1.00797 + 15.9994 = 18.0153$.

2.1.2 The basic gating mechanism

Membranes of excitable cells have so-called channels or gates through which specific ions, in particular Na^+, K^+, Ca^{2+}, Cl^-can selectively pass. These gates in fact consist of proteins which can be in an open state (O, ions can pass) or a closed state (C, ions cannot pass).

The possible actions consist of switching between the two states. This is usually represented in a simple diagram like (2.1):

$$C \underset{k^-}{\overset{k^+}{\rightleftharpoons}} O \tag{2.1}$$

The positive real numbers k^+, k^- are called *rate constants* or *proportionality constants*. The meaning of k^+ is as follows. If the concentration of closed states is given by [C], and Δt is a brief time interval, then after time Δt, the part $k^+[\mathrm{C}]\Delta t$ will have opened. Similarly, if the concentration of open states is given by [O], then after time Δt, the part $k^-[\mathrm{O}]\Delta t$ will have closed.

This is a special instance of the principle that is (inappropriately) called the *law of mass action* which states that the rate of a (chemical) process is proportional to the product of the concentrations of the components of the process. At the level of individual molecules, this reduces to a probability but usually one is only interested in the effect on a large number of molecules.

In the present situation only one component is involved. The quantity $J_+ = k^+[\mathrm{C}]$ is called the rate of transition from state C to state O. Similarly, $J_- = k^-[\mathrm{O}]$ is called the rate of transition from state O to state C.

In the present situation only the interconversion of channel states is involved. If N_C and N_O denote the numbers of cells in the closed state, respectively in the open state, then the quantity $N = N_C + N_O$ is necessarily fixed. Let us call $f_C = \frac{N_C}{N}$ the fraction of closed states and $f_O = \frac{N_O}{N}$ the fraction of open states. Then $f_O = 1 - f_C$ and we need effectively only an equation for one of f_C, f_O.

The rate of transition from state O to C is now given by $j_- = k^- f_O$ and the rate of change from state C to O is given by $j_+ = k^+(1 - f_O)$. The difference between these two rates represents the change in f_O over time:

$$\frac{df_O}{dt} = j_+ - j_- = k^+(1 - f_O) - k^- f_O = -(k^- + k^+)(f_O - \frac{k^+}{k^- + k^+}) \tag{2.2}$$

Now let us denote $\tau = \frac{1}{k^- + k^+}$ and $f_\infty = \frac{k^+}{k^- + k^+}$. Both constants are strictly positive and f_∞ is in $]0, 1[$. We then have

$$\frac{df_O}{dt} = \frac{-(f_O - f_\infty)}{\tau}. \tag{2.3}$$

We note that f_O, f_∞ are dimensionless and that τ has the dimension of time.

2.1.3 The mathematical solution of the gating equation for fixed rate constants

Equation (2.3) holds at a fixed moment in time, even if the rate constants k^+, k^- (and therefore f_∞ and τ) are time-dependent. In fact, the rate constants can depend on several external influences, for example temperature, voltage gradient (different electrical potential on the two sides of the membrane), or the presence of chemical substances that block the gates.

We now look at the relatively simple case where there are no such outside influences and k^+, k^- are constant, so that τ and f_∞ are also constant. In this case, (2.3) is a first order linear differential equation with constant coefficients. The analytical solution to such equations is well known and has the general form:

$$f_O(t) = f_\infty + Ae^{-\frac{t}{\tau}}, \tag{2.4}$$

where A is an integration constant. The solution is not unique but uniqueness is established as soon as we fix the value of $f_O(t)$ at any given time t. For example, if $f_O(0)$ is known, then we infer from (2.4) that $f_O(0) = f_\infty + A$, hence

$$f_O(t) = f_\infty + (f_O(0) - f_\infty)e^{-\frac{t}{\tau}}. \tag{2.5}$$

Obviously, every such solution converges to f_∞ as t tends to ∞ and the decay is exponential.

We note that from an experimental point of view, f_∞ can be measured if the ion transport through the membrane can be measured. Furthermore, τ can also be obtained experimentally from f_∞ and the values of any solution of (2.3) at two different times. The knowledge of f_∞ and τ then further allows to compute k^+ and k^-. It is generally hard to measure these quantities directly.

2.1.4 The Nernst potential

Biological fluids such as cytoplasm (intracellular fluid) and extracellular fluid contain many ions. These flow freely in the fluid but cannot always pass through membranes. Typically, membranes contain channels (gates) which are selectively permeable. Therefore, the concentration of an ion inside a cell may be different from the concentration outside the cell. In fact, cells often contain mechanisms that actively maintain concentration differences. This is, in particular, the task of the mitochondria which are tiny particles in cells that actively (consuming body energy) pump ions through the cell membrane.

For simplicity, let us consider a case of two monovalent ions K^+ (potassium) and Cl^- (chlorine). Monovalent means that the electrical charge of a K^+ ion is that of a proton, the charge of a Cl^- ion is that of an electron. Suppose also that we have two compartments, say *L* (left) and *R* (right) which are separated by a membrane. (One might think of *L* as being the cytoplasm of a cell and of *R* as the extracellular fluid).

First suppose that *L* has equal concentrations $[\mathrm{K}^+]_L = [\mathrm{Cl}^-]_L$ and *R* has equal concentrations $[\mathrm{K}^+]_R = [\mathrm{Cl}^-]_R$ but the concentrations in *R* are higher than the concentrations in *L*.

In that case there is no potential difference between the two membranes because the electrical charges in both *L* and *R* balance.

If we insert a non-selective pore into the cell membrane through which both K^+ and Cl^- ions can freely pass, then by diffusion the concentrations on both sides will gradually equilibrate. At the end, the potential difference between *L* and *R* will again be zero.

Suppose, however, that we insert into the membrane an ion-selective pore that allows only the passage of K^+ ions. Since $[\mathrm{K}^+]_R > [\mathrm{K}^+]_L$, the K^+ ions diffuse through the pore from *R* to *L*. Since the Cl^- ions cannot pass through the pore, each K^+ ion that passes from *R* to *L* carries a positive charge that is not counterbalanced by a Cl^- ion. Because the transfer of these charges establishes an electrical potential gradient, K^+ ions continue to move from high concentration to low concentration until the growing force due to electrical potential difference is balanced by the opposite force generated by the concentration difference.

The *equilibrium potential*, where the electrical and osmotic forces are balanced, is given by the Nernst equation. The Nernst equation is derived from the expression for the change in *Gibbs free energy* when one mol of an ion of valence z is moved across a membrane. This Gibbs free energy is given by

$$\Delta G = -RT \ln \frac{[ion]_R}{[ion]_L} + (\Delta V)Fz$$

In this expression R is the gas constant and F is Faraday's constant (see Table 2.1), T is the temperature in Kelvin (i.e. temperature in degrees Celsius plus 273.16), ΔV is the potential difference across the membrane and $[ion]_L$, $[ion]_R$ denote the ion concentrations in the fluids before and after the crossing of the membrane.

It is a physical principle that the Gibbs free energy of a system tends to decrease. Therefore the transport of ions through the pore continues as long as ΔG is negative. It stops when $\Delta G = 0$, i.e. when

$$\Delta V = \frac{RT}{zF} \ln \frac{[ion]_R}{[ion]_L} = \frac{RT \ln 10}{zF} \log_{10} \frac{[ion]_R}{[ion]_L}.$$

To get a rough idea of what this means, we consider body temperature (37 degrees Celsius) and a monovalent ion. For a 10:1 concentration ratio we get

$$\frac{RT \ln 10}{zF} = \frac{8.315(273.16 + 37)\ln(10)}{96485} = 0.0615 = 61.5\, mV.$$

Hence a ratio 10:1 in ion concentrations on the two sides of the membrane can compensate for a potential difference of 61.5 mV (millivolts). This potential difference is called the *Nernst potential*. In general, when one starts from any given concentration difference, the ions will move in such a way that the potential difference will gradually evolve to the Nernst potential.

In electrophysiology, the Nernst potential is also called the reversal potential because the departure from that point of zero current flux results in the positive or negative flow of ions. We note that this reversal potential is ion-specific, i.e. each ion has its own reversal potential which is dependent on the ion concentrations on both sides of the membrane (and to a lesser degree on the temperature). In Table 2.3 the resting ion concentrations for the K^+, Na^+ and Cl^- ions are given in two neurons. We note that the cell soma contains a large concentration of

	Cytoplasmic	Extracellular
Ion	Concentration (mM)	Concentration (mM)
Squid Giant Axon		
K$^+$	400	20
Na$^+$	50	440
Cl$^-$	40	560
Mammalian Neuron		
K$^+$	135	3
Na$^+$	18	145
Cl$^-$	7	120

Table 2.3: Resting ion concentrations. From McCormick [100].

potassium ions and a low concentration of sodium and chlorine ions, compared to the extracellular fluid.

To get a more precise understanding, suppose we have one kilogram of extracellular fluid of the squid (the solvent). How many grams of potassium ions (the solute) will be dissolved in this amount of fluid? From Table 2.3 we infer that this corresponds to 20 millimolar K$^+$ ions. Since the atomic weight of potassium is 39.102 (see §2.1.1) this amounts to $39.102 \times 0.02 = 0.78204$ grams of potassium.

2.1.5 The membrane model

We know from Ohm’s law that current flows down a voltage gradient (i.e. from a place with a higher potential to a place with a lower potential) in proportion to the resistance in the circuit. Current is therefore expressed as

$$I = \frac{V}{R} = gV. \tag{2.6}$$

where g is the inverse of R and called the *conductance*. The unit of resistance is the Ohm (Ω) and the unit of conductance is the Siemens ($S = \frac{1}{\Omega}$), cf. Table 2.2.

The conceptual idea behind contemporary electrophysiological models was formulated in the work of K.S. Cole [19], though the fundamental ideas were around before that time, e.g. in the pioneering work of A. Hodgkin and A. Huxley [73, 74].

This idea is that a cell membrane can be likened to an electronic circuit in the following sense.

1. A membrane has a phospholipid bilayer which acts as a capacitor, i.e. it accumulates ionic (electrical) charge as the electrical potential across the membrane changes. So it has a capacitance C and if we denote the potential difference across the membrane (potential on side *L* minus potential on side *R*) by V then an electrical charge of amount CV is accumulated on the *R* side.

2. The ionic permeabilities of the membrane act as resistors in an electric circuit.

3. The electrochemical driving forces for each type of channel are given by the difference between the membrane potential difference and the reversal potential of the corresponding ion; these driving forces act as batteries driving the ionic circuits.

Now let us work this out in more detail. We take the convention that the potential difference is defined as ”inner potential minus outer potential”, and that currents are positive if they move from ”outside” to ”inside”. Through the potassium K$^+$ channel a current passes which by Ohm’s law is given by

$$I_K = -g_K(V - V_K), \tag{2.7}$$

where g_K is the conductance of the K$^+$ channel; the leading ”$-$” sign is needed because of our conventions about orientation. V_K is the potassium reversal potential that is determined by the Nernst equation, and $V - V_K$ is the driving force across the membrane provided by the ionic battery.

We assume that the reversal potential for a given ion is constant; this is equivalent to assuming that there is some restorative mechanism such as ion pumps (mitochondria!) that can keep pace with electrical activity on a time scale that prevents the ionic battery from running down.

In fact, numerous ions are responsible for the electrical behaviour of a cell, and the total current is given by

$$I = \Sigma - g_i(V - V_i), \tag{2.8}$$

where the summation is over all ion currents.

In experimental situations the potential difference V is measured across the membrane and in the process an additional (”applied”) current I_{app} is introduced into the cell. Let us call $I_{cap} = I_{ion} + I_{app}$ the total (capacitive) current across the cell membrane. From the first assumption of the cell membrane model ($CV = Q$ where Q is accumulated charge) we find by derivation that

$$C\frac{dV}{dt} = I_{cap} = I_{ion} + I_{app} = -\Sigma g_i(V - V_i) + I_{app}, \tag{2.9}$$

where the summation is over all ion channels.

2.1.6 The voltage clamp

In (2.9) the conductances g_i are not constant in general. Let us denote by $\bar{g}_i$ the maximum conductance of the i-th channel, i.e. its conductance when all channels are open. Then we can write $g_i = \bar{g}_i . f_{iO}$ where f_{iO} is the fraction of open channels as in §2.1.2. In §2.1.2 we saw that f_{iO} is time-dependent and its evolution is governed by the equation

$$\frac{df_{iO}}{dt} = \frac{-(f_{iO} - f_{i\infty})}{\tau_i}, \tag{2.10}$$

in which $f_{i\infty}$ and τ_i are parameters determined by the rate constants of the ion channels. Moreover, in excitable cells the rate constants of the ion channels are not constant but depend on the potential difference across the membrane. This is easily understandable; if the electrical potential in the soma of the cell (the interior body of the cell) is, for example, higher than the potential of the surrounding fluid, then this favours the movement of positive ions out of the cell and of negative ions into it. So in fact, (2.10) should be replaced by

$$\frac{df_{iO}}{dt} = \frac{-(f_{iO} - f_{i\infty}(V))}{\tau_i(V)}, \tag{2.11}$$

The voltage clamp is an experimental method designed to measure $f_{i\infty}(V)$ and $\tau_i(V)$ directly in a given cell (it is not feasible to measure the rate constants themselves experimentally). In order to measure the voltage across a cell membrane or the current flowing through a membrane, microelectrodes are inserted into cells.

These electrodes can be used both to measure current and voltage and to apply external current. The voltage clamp is an electronic feedback device that adjusts the applied current I_{app} to match and counter the membrane currents such that the membrane voltage is held constant. To see what this accomplishes, consider a membrane with a single gated ionic current. Then (2.9) reduces to

$$C\frac{dV}{dt} = I_{cap} = I_{ion} + I_{app} = -\bar{g}f_O(t)(V - V_{rev}) + I_{app}, \tag{2.12}$$

where V_{rev} is the reversal potential given by the Nernst equation. If we can hold V constant, i.e. $\frac{dV}{dt} = 0$ by applying a current that is equal and opposite to the current flowing through the membrane, then we must have

$$I_{app} = \bar{g}f_O(t)(V - V_{rev})$$

From this type of experiments it is possible to obtain V_{rev}, $\bar{g}$ and $f_O(t)$ from which eventually $f_\infty(V)$ and $\tau(V)$ can be deduced.

To carry out a voltage clamp experiment like this, it is necessary to block all but a single type of current. While this sounds difficult and in fact is not always possible, specific toxins and pharmacological agents have been used successfully.

Also, it is not uncommon in nature. For example, tetrodotoxin (TTX) from the puffer fish selectively blocks voltage gated Na^+ - channels. Similarly, the toxin of the black mamba (an African snake) selectively blocks the K^+ - channels. The K^+- channels can also be blocked by injecting a large amount of K^+ ions into the body liquid (normally K^+ is present in high concentration in the cells and in low concentrations outside). Such injections have been used in the USA to execute criminals convicted of murder [41].

In the future we will drop the overbar in $\bar{g}$ so that g will actually denote the maximum conductance which has to be scaled by gating variables of the form f_O.

To carry out or to simulate a voltage clamp experiment, a consistent set of electrical units must be used. Here we discuss the set used in [47]. The standard unit for membrane potentials is millivolts (mV), and the characteristic times for voltage-dependent gates τ are in milliseconds. Currents are typically expressed in $\mu\mathrm{A}/\mathrm{cm}^2$ and capacitances in $\mu\mathrm{F}/\mathrm{cm}^2$. For a typical cell of area $10^{-6}\mathrm{cm}^2$, this translates to whole-cell current of picoampères (1 pA=10^{-12}A) and a whole-cell capacitance of picofarads (1 pF=10^{-12}F). This set is summarized in Table 2.4.

Name (Symbol)	Units	Abbreviation
voltage (V)	10^{-3} volt	mV
time (t)	10^{-3} second	ms
conductance (g)	10^{-9} siemens	nS
capacitance (C)	10^{-12} farad	pF
current (I)	10^{-12} ampère	pA

Table 2.4: Consistent Electrical Units.

We note that cellular dimensions are usually reported in micrometers (”microns”). Most biological channels have a conductance g on the order of 1 to 150 pS; whole-cell conductances are usually expressed in mS/cm^2.

2.2 Main examples

Over the years, many different neural models have been developed. They differ primarily in ionic currents and parameters, which are very dependent on both individual neuron and animal species.

What follows in this section, is a description of some of these models that are well-known and often used as starting point to create new, more type-specific, models. These are also the models that are used regularly as examples throughout this thesis.

2.2.1 The Hodgkin-Huxley model

Based on experiments in giant squid axons (*Loligo pealei*, also called the Atlantic squid), Hodgkin and Huxley built the first quantitative model of the electrical excitability of neurons. In a series of papers published in 1952, A.L. Hodgkin and A.F. Huxley presented the results of a series of experiments in which they investigated the flow of electric current through the surface membrane of the giant nerve fibre of a squid. In the summary paper of the Hodgkin and Huxley model [74], the authors developed a mathematical description of the behaviour of the membrane based upon these experiments, which accounts for the conduction and excitation of the fibre. The form of this description has been used as the basis for almost all other ionic current models of excitable tissues since. For this work on the basis of ’nerve action potentials’, they received the Nobel Prize in Physiology or Medicine in 1963.

The model describes three ionic currents: an inward sodium current, an outward potassium current, and a leak current. Hodgkin and Huxley discovered empirically that the conductances of the currents were not constant, but rather functions of the membrane potential and this voltage dependence was the key to understanding action potentials or spikes.

The Hodgkin-Huxley model is defined by the following equations:

$$\begin{aligned} C\frac{dV}{dt} = I - g_{Na}m^3h(V &-& V_{Na}) - g_K n^4(V - V_K) - g_L(V - V_L) \\ \frac{dm}{dt} &=& \phi\,((1-m)\,\alpha_m - m\,\beta_m); \\ \frac{dh}{dt} &=& \phi\,((1-h)\,\alpha_h - h\,\beta_h); \\ \frac{dn}{dt} &=& \phi\,((1-n)\,\alpha_n - n\,\beta_n); \end{aligned} \tag{2.13}$$

$$\begin{array}{rcll} \psi_{\alpha_m} &=& \frac{25-V}{10} & \alpha_m = \frac{\psi_{\alpha_m}}{\exp[\psi_{\alpha_m}]-1} \\ & & & \beta_m = 4\ \exp[-V/18] \\ \\ \phi &=& 3^{\frac{T-6.3}{10}} & \alpha_h = 0.07\ \exp[-V/20] \\ & & & \beta_h = \frac{1}{1+\exp[(30-V)/10]} \\ \\ \psi_{\alpha_n} &=& \frac{10-V}{10} & \alpha_n = 0.1\ \frac{\psi_{\alpha_n}}{\exp[\psi_{\alpha_n}]-1} \\ & & & \beta_n = 0.125\ \exp[-V/80] \end{array}$$

In this work the parameters $C = 1$, $g_{Na} = 120$, $V_{Na} = 115$, $g_K = 36$, $V_K = -12$, $g_l = 0.3$, $V_l = 10.559$ and $T = 6.3$ are fixed. I is varied according to the tests.

The variables m, h and n represent the change in Na channel activation (opening of channels), in Na channel inactivation (closing of channels) and in K channel activation, respectively.

2.2.2 The Connor model

In 1977, J.A. Connor et al. published a modification of the Hodgkin-Huxley model [20]. They modified the equations to approximate voltage clamp data from crustacean walking leg axons (*Callinectes sapidus* and *Cancer magister*). A new potassium current was added, with both an activation and an inactivation component, referred to as the A-current. There were also significant changes in the sodium conductance system, compared to the Hodgkin-Huxley equations, although not so pronounced that an extra current was needed.

The Connor model is defined by the following equations:

$$\begin{aligned} C\frac{dV}{dt} &= I - g_L(V - V_L) - g_{Na}m^3h(V - V_{Na}) - g_K n^4(V - V_K) \\ &\quad -g_A A^3 B(V - V_A) \end{aligned} \tag{2.14}$$

$$\begin{aligned} \frac{dm}{dt} &= \frac{m_\infty(V) - m}{\tau_m(V)} \\ \frac{dh}{dt} &= \frac{h_\infty(V) - h}{\tau_h(V)} \\ \frac{dn}{dt} &= \frac{n_\infty(V) - n}{\tau_n(V)} \\ \frac{dA}{dt} &= \frac{A_\infty(V) - A}{\tau_A(V)} \\ \frac{dB}{dt} &= \frac{B_\infty(V) - B}{\tau_B(V)} \end{aligned}$$

$$
\begin{array}{rclrcl}
\alpha_m & = & \frac{0.1(V+29.7)}{1-\exp(-0.1(V+29.7))} & \alpha_h & = & 0.07\ \exp[-0.05(V+48)] \\
\beta_m & = & 4\ \exp[-(V+54.7)/18] & \beta_h & = & \frac{1}{1+\exp[-0.1(V+18)]} \\
m_\infty & = & \frac{\alpha_m}{\alpha_m+\beta_m} & h_\infty & = & \frac{\alpha_h}{\alpha_h+\beta_h} \\
\tau_m & = & \frac{1}{3.8}\ \frac{1}{\alpha_m+\beta_m} & \tau_h & = & \frac{1}{3.8}\ \frac{1}{\alpha_h+\beta_h}
\end{array}
$$

$$
\begin{array}{rclrcl}
\alpha_n & = & 0.01\frac{V+45.7}{1-\exp[-0.1(V+45.7)]} & A_\infty & = & \left(0.0761\frac{\exp[(V+94.22)/31.84]}{1+\exp[(V+1.17)/28.93)]}\right)^{\frac{1}{3}} \\
\beta_n & = & \frac{1}{8}\ \exp(-\frac{1}{80}(V+55.7)) & \tau_A & = & 0.3632+\frac{1.158}{1+\exp[(V+55.96)/20.12]} \\
n_\infty & = & \frac{\alpha_n}{\alpha_n+\beta_n} & B_\infty & = & \frac{1}{(1+\exp[(V+53.3)/14.54])^4} \\
\tau_n & = & \frac{2}{3.8}\ \frac{1}{\alpha_n+\beta_n} & \tau_B & = & 1.24+\frac{2.678}{1+\exp[(V+50)/16.027]}
\end{array}
$$

In this study the parameters $C = 1$, $g_L = 0.3$, $E_L = -17$, $g_{Na} = 120$, $g_A = -47.7$, $V_{Na} = 55$, $g_K = 20$, $V_K = -72$ and $V_A = -75$ are fixed. I is varied according to the tests.

2.2.3 The Morris-Lecar model

In 1981, C. Morris and H. Lecar developed a neural model for muscle fibers of the barnacle (*Balanus nubilus*) [101]. The barnacle was dissected, and individual fibers from its muscles were used in doing voltage clamp measurements. Experimental work indicated that the giant barnacle muscle fiber contains primarily voltage gated K$^+$ and Ca^{2+} currents, along with a Ca^{2+}-dependent K$^+$ current. The Morris-Lecar model involves only a fast activating Ca^{2+} current, a delayed rectifier K$^+$ current and a passive leak current.

The original Morris-Lecar model has three state variables, and is defined by the following equations:

$$
\begin{array}{rcl}
C\frac{dV}{dt} & = & I_{ext} - g_L(V-V_L) - g_{Ca}M(V-V_{Ca}) - g_K N(V-V_K) \\
\frac{dM}{dt} & = & \tau_M(M_\infty - M) \\
\frac{dN}{dt} & = & \tau_N(N_\infty - N)
\end{array}
\qquad (2.15)
$$

$$\begin{aligned} M_\infty &= \tfrac{1}{2}(1+\tanh(\tfrac{V-V_1}{V_2})) & N_\infty &= \tfrac{1}{2}(1+\tanh(\tfrac{V-V_3}{V_4})) \\ \tau_M &= \phi_M \cosh(\tfrac{V-V_1}{2V_2}) & \tau_N &= \phi_N \cosh(\tfrac{V-V_3}{2V_4}) \end{aligned}$$

Later, a simplification of this model was constructed, reducing it to a 2-dimensional dynamical system:

$$\begin{aligned} C\frac{dV}{dt} &= I_{ext} - g_L(V-V_L) - g_{Ca}M_\infty(V-V_{Ca}) - g_K N(V-V_K) \\ \frac{dN}{dt} &= \tau_N(N_\infty - N) \end{aligned} \tag{2.16}$$

$$\begin{aligned} M_\infty &= \tfrac{1}{2}(1+\tanh(\tfrac{V-V_1}{V_2})) & \tau_N &= \phi\cosh(\tfrac{V-V_3}{2V_4}) \\ N_\infty &= \tfrac{1}{2}(1+\tanh(\tfrac{V-V_3}{V_4})) \end{aligned}$$

In our experiments, we mostly used this 2-dimensional version, with fixed parameters $C = 5$, $g_L = 2$, $V_L = -60$, $g_{Ca} = 4$, $V_{Ca} = 120$, $g_K = 8$, $V_K = -80$, $\phi = \frac{1}{15}$, $V_1 = -1.2$, $V_2 = 18$ and $V_4 = 17.4$. Parameters I and V_3 are usually variable, depending on the test.

We note the factor $2V_4$ in the denominator of the equation defining N_∞ in the 2-dimensional equations. These equations are the simplified Morris-Lecar equations as commonly used in scientific literature. But in fact, a derivation from first principles done by Joel Keizer and reported in [47], Chapter 2, indicates that a factor V_4 would be better justified. However, this change has little influence on the behaviour of the model, especially in the biologically relevant parameter range.

2.2.4 The Ermentrout model

As final important example we wish to list the canonical model developed by G.B. Ermentrout et al. [45], adapted from Traub and Miles [126].

$$\begin{aligned} C\frac{dV}{dt} &= I - g_{Na}m^3h(V-V_{Na}) - (g_K n^4 + g_m w + \\ &\quad g_{ahp}\frac{ca}{ca+1})(V-V_K) - g_L(V-V_L) - i_{ca} \end{aligned} \tag{2.17}$$

$$\begin{aligned}
\frac{dm}{dt} &= ((1-m)\,\alpha_m - m\,\beta_m);\\
\frac{dh}{dt} &= ((1-h)\,\alpha_h - h\,\beta_h);\\
\frac{dn}{dt} &= ((1-n)\,\alpha_n - n\,\beta_n);\\
\frac{dw}{dt} &= \frac{w_\infty - w}{t_w};\\
\frac{dca}{dt} &= -0.002 i_{ca} - \frac{ca}{80};
\end{aligned}$$

$$\begin{array}{rclrcl}
i_{ca} & = & g_{ca} m_{l,\infty}(V - V_{ca}) & m_{l,\infty} & = & \frac{1}{1+\exp[-(V+25)/2.5]} \\
\\
\alpha_m & = & \frac{0.32(V+54)}{1-\exp[-(V+54)/4]} & \alpha_h & = & 0.128\ \exp[-\frac{V+50}{18}] \\
\beta_m & = & \frac{0.28(V+27)}{\exp[(V+27)/5]-1} & \beta_h & = & \frac{4}{1+\exp[-(V+27)/5]} \\
\\
\alpha_n & = & \frac{0.032(V+52)}{1-\exp[-(V+52)/5]} & t_w & = & \frac{100}{3.3\ \exp[(V+35)/20]+exp[-(V+35)/20]} \\
\beta_n & = & 0.5\ \exp[-\frac{V+57}{40}] & w_\infty & = & \frac{1}{1+exp[-(V+35)/10]}
\end{array}$$

In this work the parameters $V_K = -100$, $V_{Na} = 50$, $V_l = -67$, $V_{ca} = 120$, $g_l = 0.2$, $g_K = 80$, $g_{Na} = 100$, $C = 1$, $g_{ca} = 1$ and $g_{ahp} = 0$ are fixed. g_m and I are varied, according to the tests.

2.3 A Classification of neural models

2.3.1 Classification by spiking behaviour

In 1948, even before introducing the famous model, Hodgkin [72] reported on different kinds of behaviour that were observed in excitable axons, more precisely in the ways in which spiking was initiated. He classified the axons into two major distinct classes: Class I and Class II.

Neurons of Class I were defined by Hodgkin as those neurons for which the frequency of the response varies smoothly with the applied current over a very broad range and which have the ability to fire at arbitrarily low frequency at the onset of firing.

In Class II neurons, the period is relatively insensitive to changes in the applied current. Also, at a constant strength of current there is either a train of impulses of fairly high frequency, or no propagated action potentials at all. Axons (or neurons) of this class are not capable of giving low frequency discharges.

Hodgkin [72] does mention however, that ’the classification is in any case of an arbitrary nature since there are transitional stages between the various groups’.

2.3.2 Classification by excitability

Rinzel and Ermentrout [110] introduced a new classification of neural models: Type I and Type II.

Type I neurons go from steady state to oscillatory behaviour through a fold bifurcation. For this type, so they claimed, repetitive firing first appears with zero frequency (homoclinic-to-saddle-node bifurcation), latency may be arbitrarily long and intermediate-sized responses (in amplitude) are not possible.

In Type II neurons, the spiking is initiated through a (subcritical) Hopf bifurcation. This leads to the onset of oscillations with a well-defined, non-zero frequency and with possibly small amplitude, and latency for firing is finite.

On reviewing Hodgkin’s classification (Section 2.3.1) and their own, they concluded that both methods of classification give identical results, and are thus equivalent. Type I, with saddle-node point onset, would match Class I. Type II, with a Hopf point bifurcation, would cause an interval of bistability, and provide all characteristics of Class II. This conclusion has been taken as reference point in a large number of subsequent papers ([21], [44], [45], [129],...).

2.3.3 Classification by subcritical and supercritical behaviour

The previous classification schemes lead to some confusion if no clear distinction is made between the subcritical and supercritical behaviour of the periodic orbits. To obtain a rigorous classification we restrict to cases where there is a stable equi-

librium before the transition, stability of the equilibrium is lost in a critical transition point and stable periodic orbits exist after the transition. For the loss of stability of the equilibrium there are four generic (codimension 1) cases : fold (= saddle-node), orbit homoclinic to saddle-node, subcritical Hopf and supercritical Hopf.
A complete classification can be based on three components:

1. The loss of stability of the stable equilibrium.
2. The existence of stable periodic orbits for values of the critical parameter on the side where the stable equilibrium exists and, in the case of existence, their extinction in the sense of moving away from the transition point.
3. The extinction of the stable periodic orbits for increasing values of the critical parameter (i.e. on the side where the equilibrium has lost stability and in the sense of moving away from that point).

So we propose the following scheme:

- Types by equilibrium bifurcation
  - Type I: fold bifurcation (including orbit homoclinic to saddle-node).
  - Type II: Hopf bifurcation (subcritical or supercritical).

  The transition between the two types is a Bogdanov-Takens point.
- Groups by subcritical behaviour
  - Group I: bistability with stable periodic orbits that shrink to a Hopf point.
  - Group II: bistability with stable periodic orbits that end with a finite nonzero period.
  - Group III: bistability with stable periodic orbits whose periods eventually grow infinitely.
  - Group IV: no bistability

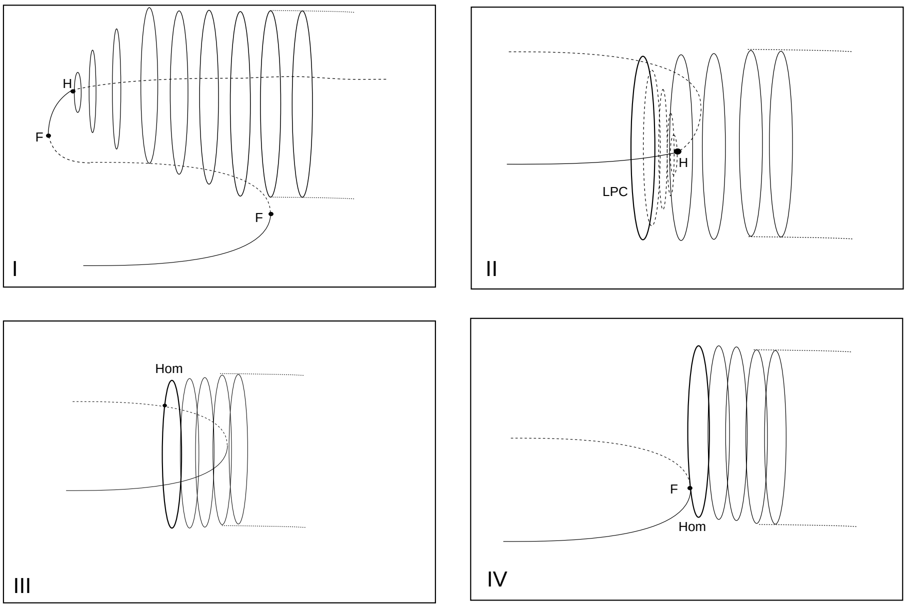


Figure 2.1: Example pictures of the 4 different Groups in subcritical behaviour. H = Hopf point; F = Fold = saddle-node; LPC = limit point of cycles; Hom = homoclinic orbit.

- Groups by supercritical behaviour
  - Group I: stable periodic orbits that shrink to a Hopf point.
  - Group II: stable periodic orbits that end with a finite nonzero period.
  - Group III: stable periodic orbits whose periods eventually grow infinitely.

In Figure 2.1 we sketch examples of the four Groups in subcritical behaviour. To classify a certain model at certain parameter values, one must state to which class it belongs in all three criteria.

The supercritical Groups I, II and III can be effectively linked to the 10 possible cases, as distinguished by Izhikevich [77], Table 4, in his study of slow-fast bursters. In this view the slow variable is replaced by the parameter as a quasistatic variable. In fact, supercritical Andronov-Hopf (in Table 4) is our Group I; fold of limit cycle, subcritical flip, subcritical Neimark-Sacker, and fold cycle on homoclinic torus (as in Table 4) are in Group II and the other bifurcations are in Group III.

It is instructive to compare our classification with the Classes of spiking systems as introduced by Izhikevich in [77], Section 3. He considers Class 1 Spiking systems, in which the oscillations terminate having arbitrarily low frequency, and Class 2 Spiking systems, in which the oscillations terminate with a non-zero frequency. In our classification the Class 1 systems are either in Group III or in Group IV by subcritical behaviour; Class 2 systems are either in Group I or Group II by subcritical behaviour.

2.4 The Phase Response Curve

2.4.1 Introduction

In the field of neural modelling, a very important concept is the phase resetting or response curve (PRC). When a neuron is firing a train of spikes (action potentials), then a short input pulse can modify the ongoing spike train. Even if the incoming pulse is purely excitatory, depending on the neuron and the exact timing of the pulse, it can either speed up or delay the next action potential. This affects properties of networks such as synchronization and phase-locking ([44], [67], ...).

Applications of the PRC to the stochastic response dynamics of weakly coupled or uncoupled neural populations can be found in [13]. They also derive PRCs for cycles close to bifurcations such as homoclinic orbits and Hopf points.

The phase response curve quantifies the linearized effect of an input pulse at a given point of the orbit on the following train of action potentials. If an input pulse does not affect the spike train, the period is unchanged and the PRC at that point of the curve is zero. If the input pulse delays the next spike, the period increases, and the PRC is negative; if the pulse speeds up the next spike, the PRC is positive.

The PRC can be used to compute the effect of any perturbing influence on the curve if the perturbation does not essentially change the dynamic pattern of the neuron; in particular it should not move the state of the neuron into a domain of attraction different from that of the spiking orbit. In the case of coupling, PRCs can be used to compute the influence of weak coupling.

In this section we define the PRC and its derivative, and give a mathematical derivation of its computation which is equivalent to the ’adjoint method’ of Ermentrout and Kopell [43]. Our method will be shown to be particularly useful in the context of the numerical continuation of orbits with a variable parameter of the system.

We also describe the computation of the response to a function. The input function f is given over the whole periodic orbit. This is more realistic in the case of coupled neurons. We further apply this functionality to a simple study of phase-locking and synchronization of phase models.

2.4.2 Response or resetting?

The terms ’phase response curve’ and ’phase resetting curve’, both abbreviated to PRC, are used interchangeably in the literature. Since there seems to be some confusion, we start with the precise definitions that will be used in the rest of this work.

Peak-based phase response curve

This is a rather intuitive and biologically oriented definition, that can easily be applied to experimental data.

The phase response curve is a curve which is defined over the timespan of one cycle in the spike train of a firing neuron with no extra input. It starts at the peak of a spike and extends to the peak of the following spike. At each point of the cycle, it indicates the effect on the oncoming spikes of an input pulse to the neuron at that particular time in its cycle.

If we define the period of the unperturbed cycle as T_{old}, and T_{new} is the time span between the previous spike and the next spike when the input pulse I is given at time t, the phase response curve is defined as

$$G(t, I) = \frac{T_{old} - T_{new}}{T_{old}} \; . \tag{2.18}$$

Phase resetting curve

This notion can be defined mathematically for any stable periodic orbit of a dynamical system. Let the period of the orbit be denoted as T_{old} and suppose that a point of the orbit is chosen as the base point (in a neural model, this point typically corresponds to a peak in the spike train).

The phase resetting curve is defined in the interval $[0,1]$, in which the variable ϕ is called the phase and is defined as

$$\phi = \frac{t}{T_{old}} (\mathrm{mod}\ 1) \ , \tag{2.19}$$

with $t \in [0, T_{old}]$.

Now suppose that an input pulse I is given to the oscillator when it is at the point $x(t)$. Mathematically, the pulse I could be any vector in the state space of the system. In neurobiological applications it is usually a vector in the direction of the voltage component of the state space, since that component is affected by the synaptic input from other neurons.

Since the orbit starting from $x(t) + I$ will close in on the stable limit cycle, there is exactly one point $x(t_1)$, $t_1 \in [0, T_{old}]$ which has the property that

$$d(y(t), z(t)) \to 0 \ \texttt{for} \ t \to \infty \tag{2.20}$$

`if`

$$\begin{cases} y(0) = x(t) + I \\ z(0) = x(t_1) \end{cases} .$$

Here $x(t_1)$ and $x(t) + I$ are said to be on the same isochron, cf. [64]. We then define

$$g(\phi, I) = \frac{t_1}{T_{old}} \ . \tag{2.21}$$

Universal definition of the phase response curve PRC

The definition of the peak-based response curve in (2.18) is satisfactory only if each period contains precisely one easily identifiable spike and transients can be ignored. I.e. we assume that the input only affects the first oncoming spike, and dies away after that. In this case the next spike will occur at time

$$T_{new} = (\phi + (1 - g(\phi, I)))\, T_{old}\ . \tag{2.22}$$

So according to (2.18)

$$PRC(\phi, I) = G(T_{old}\phi, I) = \frac{T_{old} - (\phi + (1 - g(\phi, I)))\, T_{old}}{T_{old}} = g(\phi, I) - \phi\ . \tag{2.23}$$

To avoid the difficulties related to transients we redefine the phase response curve by

$$PRC(\phi, I) = g(\phi, I) - \phi \tag{2.24}$$

in general. This definition is mathematically unambiguous and is reduced to the definition in (2.18) in the case of no transients.

2.4.3 The PRC as an adjoint problem

Let a neural model be defined by a system of differential equations

$$\dot{X} = f(X, \alpha)\ , \tag{2.25}$$

where $X \in \mathbb{R}^n$ represents the state variables and α is a vector of parameters. We consider a solution to (2.25) that corresponds to the periodically spiking behaviour of the neuron with period T. By rescaling time we find a periodic solution $x(\phi)$ with period 1, solution to the equations

$$\left\{ \begin{array}{l} \dot{x} - Tf(x, \alpha) = 0 \\ x(0) - x(1) = 0 \end{array} \right. . \tag{2.26}$$

We consider the periodic function $A(\phi) = f_X(x(\phi))$ with period 1. To (2.26) we associate the fundamental matrix solution $\Phi(\phi)$ of the non-autonomous linear equation:

$$\left\{ \begin{array}{l} \dot{\Phi}(\phi) - T\, A(\phi)\, \Phi(\phi) = 0 \\ \Phi(0) = I_n \end{array} \right. . \tag{2.27}$$

It is also useful to consider the equation adjoint to (2.27):

$$\begin{cases} \dot{\Psi}(\phi) + T\ A(\phi)^{\mathrm{T}}\ \Psi(\phi) = 0 \\ \Psi(0) = I_n \end{cases} . \tag{2.28}$$

From (2.27) and (2.28) it follows that

$$\begin{aligned} \Phi(\phi)^{\mathrm{T}}\ \dot{\Psi}(\phi) + T\ \Phi(\phi)^{\mathrm{T}}\ A(\phi)^{\mathrm{T}}\ \Psi(\phi) &= 0 \\ \Phi(\phi)^{\mathrm{T}}\ \dot{\Psi}(\phi) + (T\ A(\phi)\ \Phi(\phi))^{\mathrm{T}}\ \Psi(\phi) &= 0 \\ \Phi(\phi)^{\mathrm{T}}\ \dot{\Psi}(\phi) + \dot{\Phi}(\phi)^{\mathrm{T}}\ \Psi(\phi) &= 0 \\ \frac{d}{dt}(\Phi(\phi)^{\mathrm{T}}\ \Psi(\phi)) &= 0 \\ \frac{d}{dt}(\Psi(\phi)^{\mathrm{T}}\ \Phi(\phi)) &= 0 \end{aligned}$$

By the initial conditions this implies

$$\Psi(\phi)^{\mathrm{T}}\ \Phi(\phi) \equiv 1. \tag{2.29}$$

By taking derivatives of (2.26), we find that the tangent vector $v(\phi) = \dot{x}(\phi)$ satisfies

$$\begin{cases} \dot{v}(\phi) - T\ A(\phi)\ v(\phi) = 0 \\ v(0) - v(1) = 0 \end{cases} . \tag{2.30}$$

From this and (2.27) we infer

$$v(\phi) = \Phi(\phi)v(0)\ . \tag{2.31}$$

The monodromy matrix is the linearized return map of the dynamical system, and is defined by

$$M(\phi) = \Phi(\phi + 1)\Phi(\phi)^{-1}\ . \tag{2.32}$$

Now if we define

$$\Phi_1(\phi) = \Phi(\phi + 1)\ M(0)^{-1}, \tag{2.33}$$

then, because Φ is periodic, we can infer from (2.27) that

$$\dot{\Phi}_1(\phi) - T\ A(\phi)\ \Phi_1(\phi) = 0 \tag{2.34}$$

and

$$\Phi_1(0) = \Phi(1)M(0)^{-1} = I. \tag{2.35}$$

Thus, we conclude that

$$\Phi_1(\phi) = \Phi(\phi). \tag{2.36}$$

So from (2.33) and (2.36),

$$\begin{aligned} \Phi(\phi) &= \Phi(\phi + 1)\ M(0)^{-1} \\ \Phi(\phi + 1) &= \Phi(\phi)\ M(0)\ . \end{aligned} \tag{2.37}$$

From (2.32) and (2.37), we find

$$\Phi(\phi + 1) = \Phi(\phi)\ M(0)\ = M(\phi)\ \Phi(\phi). \tag{2.38}$$

Hence

$$M(\phi) = \Phi(\phi)M(0)\Phi(\phi)^{-1}\ . \tag{2.39}$$

The eigenvalues of $M(\phi)$ are called multipliers; the similarity (2.39) implies that they are independent of ϕ. $M(\phi)$ always has a multiplier equal to $+1$. By (2.30) and (2.31), $v(0) = v(1)$ is a right eigenvector for $M(0) = \Phi(1)$ for the eigenvalue 1. By (2.39), $v(\phi)$ is a right eigenvector of $M(\phi)$ for the eigenvalue 1 for all ϕ.

Let us assume that the limit cycle is stable, such that all multipliers different from 1 have modulus strictly less than 1. Then in particular $M(0)$ has a unique left eigenvector v_{l0} for the multiplier 1, for which $v_{l0}^{\mathrm{T}}v(0) = 1$. For all ϕ we define

$$v_l(\phi) = \Psi(\phi)v_{l0}\ . \tag{2.40}$$

It is now easy to see (from (2.28)) that

$$\left\{ \begin{array}{l} \dot{v}_l(\phi) + T\ A(\phi)^{\mathrm{T}}\ v_l(\phi) = 0 \\ v_l(0) - v_l(1) = 0 \end{array} \right. . \tag{2.41}$$

Also, for all ϕ, $v_l(\phi)$ is a left eigenvector of $M(\phi)$ for the eigenvalue 1 and

$$v_l(\phi)^{\mathrm{T}}v(\phi) = 1\ . \tag{2.42}$$

Let $R(\phi)$ be the joint right $(n-1)$-dimensional eigenspace of $M(\phi)$ that corresponds to all multipliers different from 1, i.e. $R(\phi)$ is the space orthogonal to $v_l(\phi)$.
Now, let I be a pulse given at time ϕ. We can decompose this pulse uniquely as

$$I = I_v + I_r \ , \tag{2.43}$$

with

$$\begin{cases} I_v = c\, v(\phi) \\ I_r \in R(\phi) \end{cases} , \tag{2.44}$$

where $c \in \mathbb{R}$.

Linearizing this perturbation of $x(\phi)$, we find that the effect of I_r will die out, while the effect of I_v will be to move the system vector along the orbit. The amount of the change in time is equal to c as defined in (2.44). The linearized PRC, which for simplicity we just call the PRC, at ϕ for pulse I is therefore (note that we have rescaled to period 1):

$$PRC(\phi, I) = c \ . \tag{2.45}$$

Since $v_l(\phi)$ is orthogonal to I_r we get from (2.43) and (2.42) that

$$v_l(\phi)^{\mathrm{T}}\, I \ = \ v_l(\phi)^{\mathrm{T}}\, I_v \ = c \ , \tag{2.46}$$

so

$$PRC(\phi, I) = v_l(\phi)^{\mathrm{T}} I \ . \tag{2.47}$$

The solution to the adjoint equation of (2.25), as used by Ermentrout (1996), is defined by the system

$$\begin{cases} \dot{Z}(t) = -A(t)^{\mathrm{T}}\, Z(t) \\ \frac{1}{T}\int_0^T Z(t)^{\mathrm{T}}\, \dot{X}(t) dt = 1 \end{cases} ,$$

and the periodicity condition $Z(T) = Z(0)$. It is related to our solution by the equation

$$Z(t) = T\, v_l(\frac{t}{T}) \ , \tag{2.48}$$

where $t \in [0, T]$.

Another mathematically equivalent method, though in a completely different setting, was introduced by Demir et al. [28]. They used the eigenvectors of the monodromy matrix, as we did, to compute the Perturbation Projection Vector or PPV, which is equivalent to the PRC in neural modelling.

In the case of the unscaled system (2.25), if a pulse I is given at time $t = T\phi$, then we find that

$$PRC(\phi, I) = \; v_l(\phi)^{\mathrm{T}} I \; . \tag{2.49}$$

If I is the unit vector along the first (voltage) component and we have

$$PRC(\phi, I) = \; (v_l(\phi))_1 \; , \tag{2.50}$$

where $(.)_1$ denotes the first component. This situation is so common in neural models that $(v_l(\phi))_1$ is sometimes also referred to as the phase response curve.

2.4.4 Response to a function

Instead of giving an impulse I at a fixed time $t = T\phi$ we can also add a (small) vector function $g(\phi)$, g continuous over $[0, 1]$, to the right-hand side of (2.25) to model the ongoing input from other neurons. The phase response (PR) to this ongoing input is then given by

$$PR(g) = \int_0^1 v_l(\phi)^{\mathrm{T}} g(\phi) d\phi \; . \tag{2.51}$$

We will now briefly show how the computation of the phase response to a function is done in a classical and well understood situation, namely the phase dynamics of two coupled Poincaré oscillators.

The Poincaré oscillator has been used many times as a model of biological oscillations, e.g. [52], [53]. This oscillator is most often described in a radial coordinate system (r,ϕ), where r is the distance from the origin and ϕ is the angular coordinate. The equations are

$$\left\{\begin{array}{l} \frac{dr}{dt} = k\, r\, (1 - r) \\ \frac{d\phi}{dt} = 2\, \pi \end{array}\right. \tag{2.52}$$

where k is a positive parameter. Starting at any point, except for $r = 0$, there is an evolution towards a stable attracting limit cycle at $r = 1$, which has period 1. k determines the rate of convergence.

This limit cycle can also be considered in an orthogonal coordinate system (x,y):

$$\left\{\begin{array}{l} \frac{dx}{dt} = k\, x\, (1 - \sqrt{x^2 + y^2}) - 2\, \pi\, y \\ \frac{dy}{dt} = k\, y\, (1 - \sqrt{x^2 + y^2}) + 2\, \pi\, x \end{array}\right. \tag{2.53}$$

In this coordinate system, a stable periodic orbit exists, for $t \in [0, 1]$:

$$\left\{\begin{array}{l} x = \cos(2\, \pi\, t) \\ y = \sin(2\, \pi\, t) \end{array}\right. \tag{2.54}$$

The dynamics of weakly coupled oscillators can be reduced to their phase dynamics, cf. [66] where further references can be found. We restrict to the simple setting discussed in [46]. Consider two identical, weakly coupled oscillators, with autonomous period T:

$$\left\{\begin{array}{l} X_1' = F(X_1) + \epsilon G_1(X_2, X_1) \\ X_2' = F(X_2) + \epsilon G_2(X_1, X_2) \end{array}\right.$$

with G_1 and G_2 two possibly different coupling functions, and ϵ a small positive number. Let $X_0(t)$ be an asymptotically stable periodic solution of $X' = F(X)$ with period T. Then, for ϵ sufficiently small,

$$X_j(t) = X_0(\theta_j) + O(\epsilon) \quad (j = 1, 2) \tag{2.55}$$

with

$$\theta_1' = 1 + \epsilon H_1(\theta_2 - \theta_1)$$

$$\theta_2' = 1 + \epsilon H_2(\theta_1 - \theta_2)$$

where H_j are T-periodic functions given by

$$H_j(\psi) \equiv \frac{1}{T}\int_0^T Z(t)^{\mathrm{T}}\, G_j[X_0(t+\psi), X_0(t)]dt\,, \tag{2.56}$$

where Z is the adjoint solution as defined in (2.48).

Now consider two identical Poincaré oscillators. A natural choice for the coupling functions G_j is

$$G_1(X_i, X_j) = G_2(X_i, X_j) = X_i - X_j\,.$$

It follows that, for $j = 1, 2$

$$\begin{aligned}
H_j(\phi) &= \int_0^1 v_l(t)^{\mathrm{T}}[X_0(t+\phi) - X_0(t)]dt \\
&= \int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} \cos(2\pi(t+\phi)) - \cos(2\pi t) \\ \sin(2\pi(t+\phi)) - \sin(2\pi t) \end{bmatrix} dt \\
&= \int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} \cos(2\pi t)\cos(2\pi\phi) - \sin(2\pi t)\sin(2\pi\phi) - \cos(2\pi t) \\ \sin(2\pi t)\cos(2\pi\phi) + \cos(2\pi t)\sin(2\pi\phi) - \sin(2\pi t) \end{bmatrix} dt \\
&= \cos(2\pi\phi)\int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} \cos(2\pi t) \\ \sin(2\pi t) \end{bmatrix} dt + \\
&\quad \sin(2\pi\phi)\int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} -\sin(2\pi t) \\ \cos(2\pi t) \end{bmatrix} dt - \int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} \cos(2\pi t) \\ \sin(2\pi t) \end{bmatrix} dt\,.
\end{aligned}$$

One now finds that

$$\begin{aligned}
(\theta_2 - \theta_1)' &= \epsilon(H_2(\theta_1 - \theta_2) - H_1(\theta_2 - \theta_1)) \\
&= \epsilon(H_2(\theta_1 - \theta_2) - H_2(-(\theta_1 - \theta_2))) \\
&= 2\,\epsilon\,\sin[2\pi(\theta_1 - \theta_2)]\int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} -\sin(2\pi t) \\ \cos(2\pi t) \end{bmatrix} dt\,.
\end{aligned} \tag{2.57}$$

We set $\zeta = \theta_2 - \theta_1$. For the two oscillators to be in a phase-locked state, we want the right-hand side of (2.57) to be zero. The constant function $\zeta = \frac{1}{2}$ is a solution to that; it corresponds to anti-phase phase-locking. On the other hand, if at the starting time $\zeta \neq \frac{1}{2} \pmod 1$, then we can write

$$\frac{d\zeta}{dt} = -2\,\epsilon\,\sin(2\pi\zeta)\,\alpha \tag{2.58}$$

where

$$\alpha = \int_0^1 v_l(t)^{\mathrm{T}} \begin{bmatrix} -\sin(2\pi t) \\ \cos(2\pi t) \end{bmatrix} dt\ . \tag{2.59}$$

So

$$\begin{aligned} \frac{d\zeta}{\sin(2\pi\zeta)} &= -2\,\epsilon\,\alpha dt \\ \frac{\ln\left(\frac{1-\cos(2\pi\zeta)}{\sin(2\pi\zeta)}\right)}{2\pi} &= -2\,\epsilon\,\alpha\,t + C_1\ , \end{aligned} \tag{2.60}$$

for some constant C_1. Note that

$$\begin{aligned} \frac{1-\cos(2\pi\zeta)}{\sin(2\pi\zeta)} &= \frac{1-\cos^2(\pi\zeta)+\sin^2(\pi\zeta)}{2\sin(\pi\zeta)\cos(\pi\zeta)} \\ &= \frac{\sin(\pi\zeta)}{\cos(\pi\zeta)} \\ &= \tan(\pi\zeta)\ . \end{aligned} \tag{2.61}$$

So from (2.60) and (2.61)

$$\begin{aligned} \ln\left(\frac{1-\cos(2\pi\zeta)}{\sin(2\pi\zeta)}\right) &= -4\pi\,\epsilon\,\alpha\,t + C_2 \\ \ln(\tan(\pi\zeta)) &= -4\pi\,\epsilon\,\alpha\,t + C_2 \\ \tan(\pi\zeta) &= C\ \exp(-4\pi\,\epsilon\,\alpha\,t)\ , \end{aligned} \tag{2.62}$$

for some constant $C \neq 0$.

This leads to the following conclusion about the behaviour of ζ for $t \to \infty$:

$$\begin{array}{l} \alpha > 0 \Rightarrow \zeta \to 0/1 \\ \alpha < 0 \Rightarrow \zeta \to \frac{1}{2} \\ \alpha = 0 \Rightarrow \zeta \to \texttt{constant}\ . \end{array} \tag{2.63}$$

In the first case, the two oscillators converge to in-phase phase-locking or synchronization, in the second case they move towards anti-phase phase-locking, and in the third case, the oscillators remain in the out-of-phase phase-locked state they started with.

Now α is nothing but the phase response to a function, and can be computed. In this particular case, using coupling function $G(X_2, X_1) = X_2 - X_1$, we find that $\alpha > 0$; so two identical Poincaré oscillators coupled in both directions by this function always converge to synchronization, except when they start in perfect anti-phase phase-locking, in which case they will remain in that state. This is easily checked by numerical simulation (and can be shown analytically).

Analogously, it is easy to compute α for other coupling functions, and to check the result by simulations. E.g. if $G_1(X_2, X_1) = G_2(X_2, X_1) = X_1 - X_2$, then $\alpha < 0$, so the two Poincaré oscillators always converge to anti-phase phase-locking.

A more interesting case is obtained if we set

$$\begin{aligned} X_1 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, X_2 = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}, \\ G_1(X_2, X_1) = \begin{pmatrix} x_3 - x_1 \\ 0 \end{pmatrix}, G_2(X_1, X_2) = \begin{pmatrix} x_1 - x_3 \\ 0 \end{pmatrix}. \end{aligned} \tag{2.64}$$

In this case we again find $\alpha > 0$ and there is always synchronization, except for when we start in exact anti-phase. If we change the sign of the coupling in (2.64), to obtain the function

$$G_1(X_2, X_1) = \begin{pmatrix} x_1 - x_3 \\ 0 \end{pmatrix}, G_2(X_1, X_2) = \begin{pmatrix} x_3 - x_1 \\ 0 \end{pmatrix}, \tag{2.65}$$

then α also changes sign, and there is always anti-phase locking (except when starting from perfect synchrony). These two are illustrated by numerical simulations in Figure 2.2.

When we set

$$G_1(X_2, X_1) = \begin{pmatrix} x_4 - x_2 \\ 0 \end{pmatrix}, G_2(X_1, X_2) = \begin{pmatrix} x_2 - x_4 \\ 0 \end{pmatrix}, \tag{2.66}$$

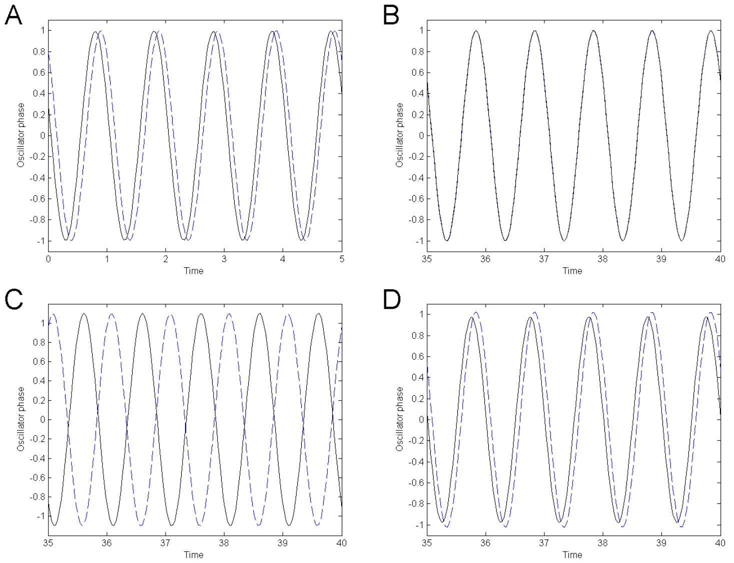


Figure 2.2: Simulation results with coupled Poincaré oscillators. A) Initial condition: 2 oscillators out of phase. B) Using coupling function (2.64), the oscillators synchronize. C) Using coupling function (2.65), the oscillators go into anti-phase. D) Using coupling function (2.66), the oscillators remain locked in their original state.

then α is zero (up to truncation and rounding errors), so the oscillators keep their initial out-of-phase phase-locking. This is confirmed by numerical simulations, as shown in Figure 2.2.

Chapter 3

New features of MatCont

MATCONT and CL_MATCONT are MATLAB numerical continuation packages for the interactive study of dynamical systems and bifurcations. The work started in 2000 and the first publications appeared in 2003. In this software we improved some aspects and added several new functionalities. Some of the new functionalities were previously only available as toolboxes that can be used by experts, e.g. continuation of homoclinic orbits. Several others were never implemented at all, such as automatic computation of phase response curves and their derivatives. After a general introduction on the software packages, one section is devoted to each of the most important modifications of the program, namely the phase response curves, homoclinic orbits and the use of C-code that is compiled at start-up time.

3.1 MATCONT and CL_MATCONT

3.1.1 Existing software

During the last decades, considerable efforts have been made to develop general-purpose software tools for bifurcation analysis of dynamical systems of the form

$$\frac{dx}{dt} = f(x,\alpha), \quad x \in \mathbb{R}^n,\ \alpha \in \mathbb{R}^p \tag{3.1}$$

(see Bibliographical Notes to Chapter 10 of [86] for references). One may distinguish at least three generations of such software:

1. *Non-interactive packages and codes* were developed in the beginning of the 1980s and written in FORTRAN. They allowed one to continue equilibria and cycles of systems of differential equations, as well as their simplest bifurcations (limit point, Hopf, period-doubling). The most widely used packages of this generation were AUTO86 and LINLBF. There were a few other programs available, which could perform normal form computations, namely: STUFF, BIFOR2, CYCLE.

2. *Interactive programs* for bifurcation analysis of ODEs appeared in the end of the 1980s, when workstations and IBM-PC compatible computers became widely available at universities and general research institutes. The first fully interactive bifurcation software was LOCBIF, that was based on LINLBF and worked on PCs under MS-DOS. It had a simple GUI with buttons, windows, and pull-down menus, characteristic for all programs of this generation. The programs AUTO94/97/2000 and XPPAUT, which ran under UNIX and used the efficient numerics of AUTO86, together with a simple GUI for X-Windows, also belong to this generation. All programs supported the on-line input of the right-hand sides of (3.1) and its compilation, either by a special built-in compiler or by calling a standard FORTRAN compiler. They allowed the user to continue equilibria, cycles, and their codimension-one bifurcations. Computed curves could be plotted in fixed graphic windows. All mentioned programs had a closed architecture.

3. The third generation includes DSTOOL and CONTENT, developed in the 1990s. These programs have features of the so-called *software environments*, meaning that the user can define/modify a dynamical model (3.1), perform its rather complete analysis, and export results of this analysis in a graphical form, all without leaving the program. The programs have an elaborate GUI and run on several platforms, including popular workstations running Motif under UNIX, and PCs running OpenMotif under Linux or just MS-Windows. They provide off- or on-line help and extensive documentation for users and developers. It is possible, though hard, to ex-

> tend them. Both programs support simulation of (3.1). DsTool computes equilibria and their codim 1 bifurcations using parts of the LINLBF-code. CONTENT, that is written in C/C++, supports the continuation of equilibria of (3.1) and their bifurcations with $\mathrm{codim} \leq 2$, as well as the continuation of cycles using AUTO-like algorithms. Moreover, it computes normal forms for many equilibrium bifurcations, using internally generated symbolic derivatives of order ≤ 3.

Computations for ODEs currently supported by the most widely used software packages AUTO97/2000 [38] and CONTENT 1.5 [85] are indicated in Table 3.1.

3.1.2 Bifurcation software MatCont

Despite all efforts, none of the above-mentioned packages covers the whole range of known bifurcations in ODEs, even with two control parameters ($p = 2$). The data exchange between existing programs is practically impossible due to individual data formats. Moreover, for some important bifurcation problems, such as normal form computations for cycle bifurcations, no robust and efficient numerical methods have been developed. None of the existing software represents the results of the analysis in a form suitable for standard control, identification, or visualization software. Therefore, none fits well into the standard engineering software environments. Existing software tools are hardly extendible, since all of them are written in relatively low-level programming languages, like FORTRAN and C.

Recognizing these deficiencies lead to a research project to develop the next generation bifurcation software in MATLAB, now known as MatCont, in which groups from Belgium and The Netherlands, as well as individual scientists from Canada, US, Switzerland, and Russia were involved. Early versions of MatCont were described in [31], [32]. MatCont also has a command-line version, called CL_MatCont.

Relationships between objects of codimension $0, 1$ and 2 computed by MatCont 2.* are presented in Figure 3.1, while the symbols and their meaning are summarized in Table 3.2, where the standard terminology is used, see [86].

Table 3.1: Supported functionalities for ODEs in AUTO (A), CONTENT (C) and MATCONT (M).

	A	C	M
time-integration		+	+
Poincaré maps			+
monitoring user functions along curves computed by continuation	+	+	+
continuation of equilibria	+	+	+
detection of branch points and codim 1 bifurcations (limit and Hopf points) of equilibria	+	+	+
computation of normal forms for codim 1 bifurcations of equilibria		+	+
continuation of codim 1 bifurcations of equilibria	+	+	+
detection of codim 2 equilibrium bifurcations (cusp, Bogdanov-Takens, fold-Hopf generalized and double Hopf)		+	+
computation of normal forms for codim 2 bifurcations of equilibria			+
continuation of codim 2 equilibrium bifurcations in three parameters		+	
continuation of limit cycles	+	+	+
computation of phase response curves and their derivatives			+
detection of branch points and codim 1 bifurcations (limit points, flip and Neimark-Sacker (torus)) of cycles	+	+	+
continuation of codim 1 bifurcations of cycles	+		+
branch switching at equilibrium and cycle bifurcations	+	+	+
continuation of branch points of equilibria and cycles			+
computation of normal forms for codim 1 bifurcations of cycles			+
detection of codim 2 bifurcations of cycles			+
continuation of orbits homoclinic to equilibria	+		+

Type of object	Label
Point	P
Orbit	O
Equilibrium	EP
Limit cycle	LC
Limit Point (fold) bifurcation	LP
Hopf bifurcation	H
Limit Point bifurcation of cycles	LPC
Neimark-Sacker (torus) bifurcation	NS
Period Doubling (flip) bifurcation	PD
Branch Point	BP
Cusp bifurcation	CP
Bogdanov-Takens bifurcation	BT
Zero-Hopf bifurcation	ZH
Double Hopf bifurcation	HH
Generalized Hopf (Bautin) bifurcation	GH
Branch Point of Cycles	BPC
Cusp bifurcation of Cycles	CPC
Generalized Period Doubling	GPD
Chenciner (generalized Neimark-Sacker) bifurcation	CH
1:1 Resonance	R1
1:2 Resonance	R2
1:3 Resonance	R3
1:4 Resonance	R4
Fold–Neimark-Sacker bifurcation	LPNS
Flip–Neimark-Sacker bifurcation	PDNS
Fold-flip	LPPD
Double Neimark-Sacker	NSNS

Table 3.2: Equilibrium- and cycle-related objects and their labels within the GUI

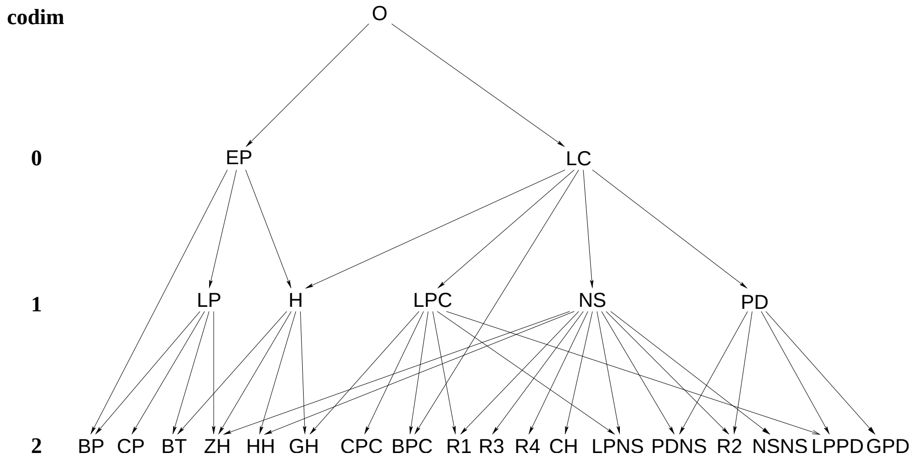


Figure 3.1: The graph of adjacency for equilibrium and limit cycle bifurcations in MatCont

An arrow in Figure 3.1 from O to EP or LC means that by starting time integration from a given point we can converge to a stable equilibrium or a stable limit cycle, respectively. In general, an arrow from an object of type A to an object of type B means that that object of type B can be detected (either automatically or by inspecting the output) during the computation of a curve of objects of type A. For example, the arrows from EP to H, LP, and BP mean that we can detect H, LP and BP during the equilibrium continuation. Moreover, for each arrow traced in the reversed direction, i.e. from B to A, there is a possibility to start the computation of the solution of type A starting from a given object B. For example, starting from a BT point, one can initialize the continuation of both LP and H curves. Of course, each object of codim 0 and 1 can be continued in one or two system parameters, respectively.

Recently, many improvements and new features were incorporated in MatCont. It is on these additions that this chapter will elaborate.

MatCont is freely available at `http://www.matcont.UGent.be` upon a registration. It requires MATLAB 6.* or higher.

3.2 Computation of the PRC

In Section 2.4.3, we have described a new algorithm to compute the PRC of a limit cycle. The method is equivalent to an existing method, the so-called adjoint method, but our implementation in MATCONT, as part of the boundary value problem for the limit cycle of the stable orbit, is faster and more robust than any existing method; in fact we get the PRC as a by-product of the computation of the limit cycle. In this section we explain the implementation of the algorithm in detail.

3.2.1 Survey of methods

Currently, two classes of methods are often used to compute PRCs.

The simplest methods are direct applications of definition (2.18). I.e. using simulations, one passes through the cycle repeatedly, each time giving an input pulse at a different time, and measuring the delay of the spike. From this a PRC curve can be computed. This was done by Guevara et al. [65] and many others.

This method has several advantages. It is simple and does not require any special software (only a good time integrator is needed). It can be used for arbitrarily large perturbations, even if the next spike would be delayed by more than the period of one cycle in which case the term ’delay’ might be confusing.

Depending on the required accuracy of the PRC this method typically takes a few seconds. In applications where the PRC of one given limit cycle is desired, it is usually quite satisfactory.

The other methods are based on the use of the equation adjoint to the dynamical system that describes the activity of the neuron. The mathematical basis of this approach goes back to Malkin [98] and [99], see also [10] and [42]. An easily accessible reference is in [75, §9.2].

The idea to use this method for numerical implementation goes back to Ermentrout and Kopell [43]. The implementation described in [44], and available through Ermentrout’s software package XPPAUT [46], is based on the backward integration of the adjoint linear equation of the dynamical system.

This method is mathematically better defined and more general than the simulation method, because it does not use the notion of a spike. It is also more restricted because it does not measure the influence of a real finite perturbation but rather a linearization of this effect which becomes more accurate if the perturbation gets smaller.

As noted in [44], the accuracy of the method based on backward integration is limited for two reasons. First, the adjoint Jacobian of the dynamical system is not available analytically. It has to be approximated by finite differences from a previously numerically computed discretized form of the orbit. Secondly, the integration of the linear equations also produces the usual numerical errors.

As noted before, the method that we propose is mathematically equivalent to the adjoint method, but the implementation is new. The main advantage is that it is much faster. Though the increase in speed is impressive, this is not very relevant if only one or a few PRCs are needed. The main application of our method therefore lies in cases where a large number of PRCs is needed, such as cases where the evolution of PRCs is important when a parameter is changed.

Another advantage is that the sources of error from the existing method do not apply to the method that we propose. In fact, we can compute PRCs even in cases where the stable periodic orbits are hard to find by direct time integration.

3.2.2 Orthogonal collocation in MATCONT

In MATCONT, as in AUTO [38] and CONTENT [85], the continuation of limit cycles is based on (2.26) and uses orthogonal collocation. We briefly summarize this technique before describing its extension to compute phase response curves. For details on the orthogonal collocation method we refer to [3] and [27]. The idea to use the discretized equation to solve the adjoint problem was used in a completely different context in [87].

Since a limit cycle is periodic, we need an additional equation to fix the phase. For this phase equation an integral condition is used. The resulting full system is the following:

$$\begin{cases} \dot{x}(t) - T\, f(x(t), \alpha) = 0 \\ x(0) - x(1) = 0 \\ \int_0^1 \dot{\tilde{x}}^{\mathrm{T}}(t) x(t) dt = 0 \end{cases}, \tag{3.2}$$

where $\tilde{x}$ is some initial guess for the solution, typically obtained from the previous continuation step. This system is referred to as the boundary value problem or BVP.

Now to describe a continuous limit cycle numerically, it needs to be stored in a discretized form. First the interval $[0, 1]$ is subdivided into N intervals:

$$0 = \tau_0 < \tau_1 < \cdots < \tau_N = 1\ . \tag{3.3}$$

The grid points $\tau_0, \tau_1, \ldots, \tau_N$ form the coarse mesh. In each interval $[\tau_i, \tau_{i+1}]$, the limit cycle is approximated by a degree m polynomial, whose values are stored in equidistant mesh points on each interval, namely,

$$\tau_{i,j} = \tau_i + \frac{j}{m}(\tau_{i+1} - \tau_i) \quad (j = 0, 1, ..., m)\ . \tag{3.4}$$

These grid points form the fine mesh. In each interval $[\tau_i, \tau_{i+1}]$ we then require the polynomials to satisfy the BVP exactly at m collocation points. The best choice for these collocation points are the Gauss points $\zeta_{i,j}$, i.e. the roots of the Legendre polynomial of degree m, relative to the interval $[\tau_i, \tau_{i+1}]$ [27].

For a given vector function $\eta \in \mathcal{C}^1([0, 1], \mathbb{R}^n)$, we consider two different discretizations and two weight forms:

- $\eta_M \in \mathbb{R}^{(Nm+1)n}$ the vector of the function values at the mesh points;
- $\eta_C \in \mathbb{R}^{Nmn}$ the vector of the function values at the collocation points;
- $\eta_{WL} \in \mathbb{R}^{(Nm+1)n}$ the vector of the function values at the mesh points, each multiplied with its coefficient for piecewise Lagrange quadrature;
- $\eta_{WG} = \begin{bmatrix} \eta_{W_1} \\ \eta_{W_2} \end{bmatrix} \in \mathbb{R}^{(Nm+1)n}$ where η_{W_1} is the vector of the function values at the collocation points multiplied by the Gauss-Legendre weights and the lengths of the corresponding mesh intervals, and $\eta_{W_2} = \eta(0)$.

To explain the use of the weight forms we first consider a scalar function $f \in \mathcal{C}^0([0,1],\mathbb{R})$. Then the integral $\int_0^1 f(t)dt$ can be numerically approximated by appropriate linear combinations of function values and this can be done in several ways.

For background on quadrature methods we refer to [30], Chapter 9. If the fine mesh points are used, then the best approximation has the form

$$\sum_{i=0}^{N-1}\sum_{j=0}^{m} l_{m,j}(f_M)_{i,j}(\tau_{i+1}-\tau_i) \tag{3.5}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{m-1}(f_{WL})_{i,j} + (f_{WL})_{N-1,m}\ . \tag{3.6}$$

In (3.5) the coefficients $l_{m,j}$ are the Lagrange quadrature coefficients and $(f_M)_{i,j} = f(\tau_{i,j})$; (3.5) is the exact integral if $f(t)$ is a piecewise polynomial of degree m or less. (3.6) is a reorganization of (3.5) and defines f_{WL}.

The integral $\int_0^1 f(t)g(t)dt$ ($f,g \in \mathcal{C}^0([0,1],\mathbb{R})$) is then approximated by the vector inner product $f_{WL}^{\mathrm{T}} g_M$. For vector functions $f,g \in \mathcal{C}^0([0,1],\mathbb{R}^n)$, the integral $\int_0^1 f(t)^{\mathrm{T}} g(t)dt$ is formally approximated by the same expression: $f_{WL}^{\mathrm{T}} g_M$.

If the collocation points are used, then the best approximation has the form

$$\sum_{i=0}^{N-1}\sum_{j=1}^{m}\omega_{m,j}(f_C)_{i,j}(\tau_{i+1}-\tau_i) = \sum_{i=0}^{N-1}\sum_{j=1}^{m}(f_{W_1})_{i,j}\ , \tag{3.7}$$

where $(f_C)_{i,j} = f(\zeta_{i,j})$ and $\omega_{m,j}$ are the Gauss-Legendre quadrature coefficients. Formula (3.7) delivers the exact integral if $f(t)$ is a piecewise polynomial of degree $2m-1$ or less.

The integral $\int_0^1 f(t)g(t)dt$ ($f,g \in \mathcal{C}^0([0,1],\mathbb{R})$) is approximated with Gauss-Legendre by $f_{W_1}^{\mathrm{T}} g_C = f_{W_1}^{\mathrm{T}} L_{C\times M} g_M$. For vector functions $f,g \in \mathcal{C}^0([0,1],\mathbb{R}^n)$, the integral $\int_0^1 f(t)^{\mathrm{T}} g(t)dt$ is formally approximated by the same expression: $f_{W_1}^{\mathrm{T}} g_C = f_{W_1}^{\mathrm{T}} L_{C\times M} g_M$.

Here we formally used the structured sparse matrix $L_{C\times M}$ that converts a vector η_M of function values at the mesh points into the vector η_C of its values at the collocation points:

$$\eta_C = L_{C\times M}\eta_M \, . \tag{3.8}$$

This matrix is usually not formed explicitly; its entries are fixed and given by the values of the Lagrange interpolation functions in the collocation points.

In the Newton steps for the computation of the solution to (3.2), we solve matrix equations with the Jacobian matrix of the discretization of (3.2):

$$\begin{bmatrix} (D - TA(t))_{C\times M} & (-f(x(t),\alpha))_C \\ (\delta_0 - \delta_1)^{\mathrm{T}}_M & 0 \\ (\dot{\tilde{x}}^{\mathrm{T}}(t))^{\mathrm{T}}_{WL} & 0 \end{bmatrix} . \tag{3.9}$$

In (3.9) the matrix $(D - TA(t))_{C\times M}$ is the discretized form of the operator $D - TA(t)$ where D is the differentiation operator. So we have $(D - TA(t))_{C\times M}\eta_M = (\dot{\eta}(t) - TA(t)\eta(t))_C$. We note that this is a large, sparse and well structured matrix. In AUTO [38] and CONTENT [85] this structure is fully exploited; in MATCONT only the sparsity is exploited by using the MATLAB sparse solvers.

We note that the evaluation of $(D - TA(t))_{C\times M}$ takes most of the computing time in the numerical continuation of limit cycles.

Furthermore, $(\delta_0 - \delta_1)^{\mathrm{T}}_M$ is the discretization in the fine mesh points of the operator $\delta_0 - \delta_1$ where δ_0, δ_1 are the Dirac evaluation operators in 0 and 1 respectively. So the (2,1)-block in (3.9) is an $(n \times (Nm + 1)n)$-matrix whose first $(n \times n)$-block is the identity matrix I_n and whose last $(n \times n)$-block is $-I_n$; all other entries are zero.

We note that in a continuation context, (3.9) is extended by an additional column, that contains $(-Tf_\alpha(x(t),\alpha))_C$ where α is the free parameter, and by an additional row, that is added by the continuation algorithm.

3.2.3 PRC implementation in MatCont

Now if the limit cycle is computed, and we want to compute the PRC, what we really need is $v_l(t)$, solution to (2.41) and (2.42). So $v_l(t)$ is defined up to scaling by the condition that it is a null vector of the operator

$$\phi_2 : \mathcal{C}^1([0,1],\mathbb{R}^n) \to \mathcal{C}^0([0,1],\mathbb{R}^n) \times \mathbb{R}^n,$$

where

$$\phi_2(\zeta) = \left[\begin{array}{c} \dot{\zeta} + TA^{\mathrm{T}}\zeta \\ \zeta(0) - \zeta(1) \end{array} \right].$$

This means that v_l is in the kernel of ϕ_2 and that $\dot{v}_l(t) + TA^{\mathrm{T}}v_l(t) = 0$ and $v_l(0) - v_l(1) = 0$.

Then for all $g \in \mathcal{C}^1([0,1],\mathbb{R}^n)$, we have

$$\begin{array}{rcl} \int_0^1 g(t)^{\mathrm{T}}\dot{v}_l(t)dt + \int_0^1 Tg(t)^{\mathrm{T}}A(t)v_l(t)dt & = & 0, \\ \Rightarrow \quad g(t)^{\mathrm{T}}v_l(t)|_0^1 - \int_0^1 \dot{g}(t)^{\mathrm{T}}v_l(t)dt + \int_0^1 Tg(t)^{\mathrm{T}}A(t)v_l(t)dt & = & 0, \\ \Rightarrow \quad g(1)^{\mathrm{T}}v_l(1) - g(0)^{\mathrm{T}}v_l(0) - \int_0^1 (\dot{g}(t) - TA(t)^{\mathrm{T}}g(t))^{\mathrm{T}}v_l(t)dt & = & 0, \\ \Rightarrow \quad -(g(0) - g(1))^{\mathrm{T}}v_l(0) - \int_0^1 (\dot{g}(t) - TA(t)^{\mathrm{T}}g(t))^{\mathrm{T}}v_l(t)dt & = & 0, \\ \Rightarrow \quad \left\langle \left[\begin{array}{c} \dot{g} - TA^{\mathrm{T}}g \\ g(0) - g(1) \end{array} \right], \left[\begin{array}{c} v_l \\ v_l(0) \end{array} \right] \right\rangle & = & 0. \end{array}$$

Since this holds for any g, we have shown that

$$\left[\begin{array}{c} v_l \\ v_l(0) \end{array} \right] \perp \phi_1(\mathcal{C}^1([0,1],\mathbb{R}^n))$$

where

$$\phi_1 : \mathcal{C}^1([0,1],\mathbb{R}^n) \to \mathcal{C}^0([0,1],\mathbb{R}^n) \times \mathbb{R}^n,$$

$$\phi_1(\zeta) = \left[\begin{array}{c} \dot{\zeta} - TA\zeta \\ \zeta(0) - \zeta(1) \end{array} \right].$$

So $\left[\begin{array}{c} v_l(t) \\ v_l(0) \end{array} \right]$ is orthogonal to the range of $\left[\begin{array}{c} D - TA(t) \\ \delta_0 - \delta_1 \end{array} \right]$. Now, by (2.42) and the fact that $v(t) = \dot{x}(t)$, we can state that

$$\left[\, ((v_l)_{WG})^{\mathrm{T}} \; 0 \, \right] \left[\begin{array}{cc} (D - TA(t))_{C\times M} & (-f(x(t),\alpha))_C \\ (\delta_0 - \delta_1)^{\mathrm{T}}_M & 0 \\ (\dot{x}^{\mathrm{T}}(t))^{\mathrm{T}}_{WL} & 0 \end{array} \right] = \left[\begin{array}{cc} 0 & -\dfrac{1}{T} \end{array} \right]. \tag{3.10}$$

The matrix in (3.10) is freely available, since it is the same one as in (3.9). Obtaining $(v_l)_{WG}$ from (3.10) is equivalent to solving a large sparse system, a task which is done efficiently by MATLAB (see also Section 3.5).

The first part $(v_l)_{W_1}$ of the obtained vector $(v_l)_{WG}$ represents $v_l(t)$, as used in equation (2.47), but it is evaluated in the collocation points, and still weighted with the interval widths and the Gauss-Legendre weights. The second part $(v_l)_{W_2}$ of the obtained vector $(v_l)_{WG}$ is equal to $v_l(0)$. In many important cases (see e.g. Section 2.4.4) $(v_l)_{W_1}$ is precisely what we need because $v_l(t)$ will be used to compute integrals of the form

$$\langle v_l, g\rangle = \int_0^1 v_l^{\mathrm{T}} g(t) dt \; ,$$

where $g(t)$ is a given vector function that is continuous in $[0, 1]$. This integral is then numerically approximated by the vector inner product

$$(v_l)_{W_1}^{\mathrm{T}} g_C = \; (v_l)_{W_1}^{\mathrm{T}} L_{C\times M} g_M \; . \tag{3.11}$$

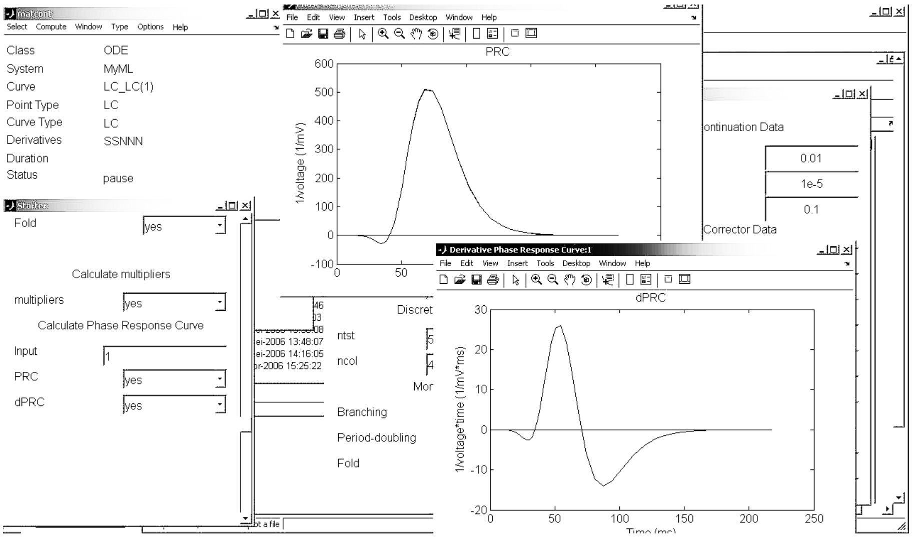
Nevertheless, in some cases (see Section 4.2) we want to know $(v_l)_M$. Since we know the Gauss-Legendre weights and the interval widths, we can eliminate them explicitly from $(v_l)_{W_1}$ and obtain $(v_l)_C$. Now (3.8) converts values in mesh points to values in collocation points. In addition, we know that $v_l(0) = v_l(1) = (v_l)_{W_2}$. Combining these results we get

$$\begin{bmatrix} (v_l)_C \\ v_l(0) \end{bmatrix} = \begin{bmatrix} L_{C\times M} \\ 0.5 \; 0 \ldots 0 \;\; 0.5 \end{bmatrix} (v_l)_M \; , \tag{3.12}$$

where $\begin{bmatrix} L_{C\times M} \\ 0.5 \; 0 \ldots 0 \;\; 0.5 \end{bmatrix}$ is sparse, square and well conditioned and so (3.12) can be solved easily by MATLAB to get $(v_l)_M$. We note that in this case (and only in this case) it is necessary to form $L_{C\times M}$ explicitly.

Now we have access to all elements needed to compute the PRC using (2.49).

The derivative phase response curve or dPRC can now easily be computed by equation (2.41). The dPRC has some very important applications. For examples



Class ODE
System MyML
Curve LC_LC(1)
Point Type LC
Curve Type LC
Derivatives SSNNN
Duration
Status pause

Fold yes
multipliers yes
Input 1
PRC yes
dPRC yes

46
03
08
ei-2006 13:48:07
ei-2006 14:16:05
or-2006 15:25:22
ot a file

ntst 5
ncol 4

0.01
1e-5
0.1

Figure 3.2: Screenshot of MatCont for computing the PRC and dPRC during LC continuation.

we refer to Section 4.2.

MatCont and CL_MatCont support the computation of the PRC and dPRC of limit cycles during continuation, using our new method. The use in MatCont is easy: before starting the actual limit cycle continuation, the user can specify in the starter window whether he wants to compute the PRC, dPRC or both, and he needs to indicate the input vector used. When a scalar is given as input, then the vector has this scalar as first entry and all other entries are zero. Then in separate plotting windows, for each computed step in limit cycle continuation, the PRC and/or dPRC are computed and plotted. An example screenshot is shown in Figure 3.2.

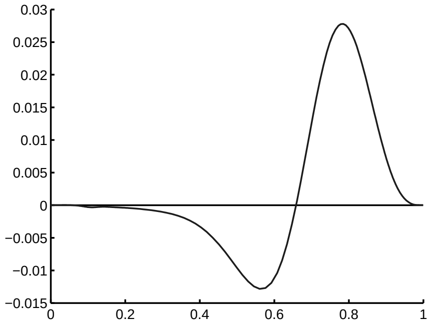


Figure 3.3: PRC of the Hodgkin-Huxley model at $I = 12$.

3.2.4 Accuracy of our method

Comparison to direct method

Here we compare our method of computing the PRC to the direct method, which consists of giving input pulses at different points in the cycle, and measuring the resulting changes in the cycle period, cf. §3.2.1. As a test system, we use the Hodgkin-Huxley system, as described in Section 2.2.1.

Figure 3.3 shows the PRC for this model for $I = 12$, as computed using our new method. The period of the limit cycle is then 13.72 seconds. During a continuation experiment, this computation takes about 0.04 seconds.

Figure 3.4 shows two PRCs for the same model and the same parameter values, that were computed in the direct way. The PRCs were computed for different pulse amplitudes and durations: pulse amplitudes are 10 and 20, and pulse durations are 0.05 and 0.15 seconds, for Figures 3.4A and 3.4B respectively.

Visually, it is clear that the shapes of the curves match. A rough computation shows that the matching is also quantitative. Indeed, the situation of Figure 3.4A corresponds to a resetting pulse of size $10 \times 0.05 = 0.5$ (millivolts). Dividing the maximal value of the computed PRC by 0.5 we obtain $0.014/0.5 = 0.028$ (per millivolt). Similarly, for the situation of Figure 3.4B we obtain $0.075/(20 \times 0.15) = 0.025$ (per millivolt). This closely matches the computed maximal value

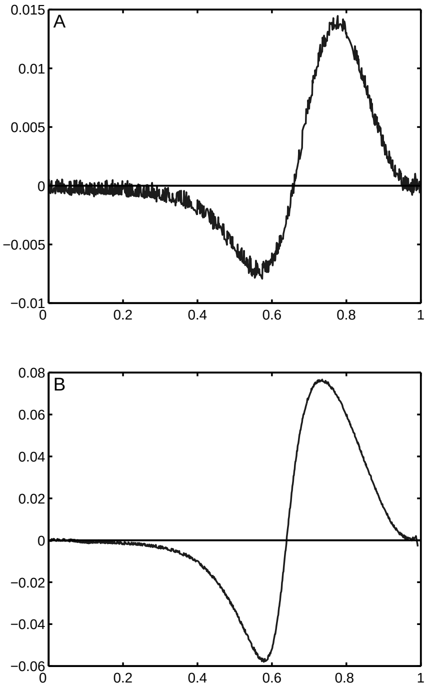


Figure 3.4: Experimentally obtained PRCs for the Hodgkin-Huxley model. A) pulse amplitude = 10, pulse duration = 0.05; B) pulse amplitude = 20, pulse duration = 0.15

in Figure 3.3.

This shows that our method is accurate and applicable, not only for infinitesimally small input pulses, but also for pulses of finite size and duration.
The direct computation of these experimental PRCs, with the limited precision they have (as is obvious from the figures), took between 60 and 70 seconds each. Also, the smaller the inputs for an experimental computation, the higher the precision has to be, since the PRC amplitudes shrink, and thus noise due to imprecision increases in relative size. This is already evident from the difference between A and B in Figure 3.4. And computation time increases with increasing precision.

Three classical situations

In this subsection we check that in some standard situations, our results correspond to those in the literature. In fact, our figures almost perfectly match the ones computed in [44], except for the scaling factor (which is the cycle period). This should not come as a surprise since in the absence of computational errors (rounding, truncation, etcetera) the used methods are equivalent. Here we present a couple of the most widely known models. The equations and fixed parameter values for all models are listed in Section 2.2.

The Hodgkin-Huxley model is known to exhibit a PRC with a positive and a negative region. So two coupled Hodgkin-Huxley neurons with excitatory connections can still slow down each others spike rate. Figure 3.3 shows the PRC for this model for $I = 12$, where the limit cycle has a period of 13.72.

The Morris-Lecar model is known to have different types of behaviour and phase response curves at different parameter values [58].

Figure 3.5A shows the PRC at $V_3 = 4$ and $I = 45$, where the cycle period is 62.38. We clearly see a negative and positive regime in the PRC. In Figure 3.5B the PRC is shown at $V_3 = 15$ and $I = 39$, where the period is 106.38; the PRC is practically non-negative.

Finally, we show a result for the Connor model, which has a non-negative PRC. Figure 3.6 depicts this PRC, for $I = 8.5$ and period 98.63.

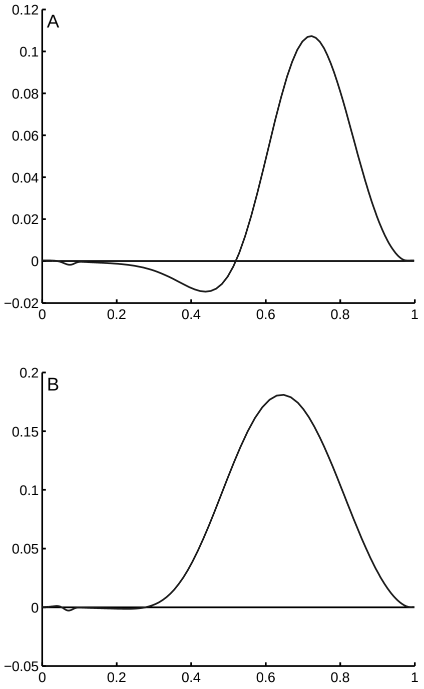


Figure 3.5: PRCs of the Morris-Lecar model. A) PRC at $V_3 = 4$ and $I = 45$. B) PRC at $V_3 = 15$ and $I = 39$.

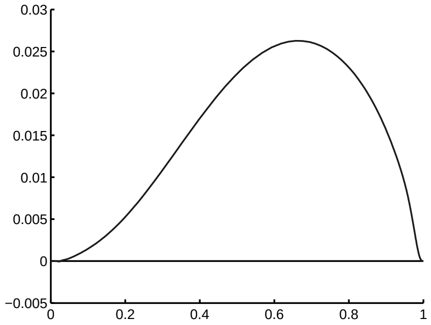


Figure 3.6: PRC of the Connor model at $I = 8.5$.

3.2.5 Robustness of our method

Our method is robust in the sense that it can compute PRCs for all limit cycles that can be found through continuation; there are no further sources of error as in the traditional implementation of the computation of the PRC, cf. [44]. In fact we can compute PRCs even for limit cycles that are hard to find by any other means than numerical continuation, which can happen when their domain of attraction is not easily found.

In the Hodgkin-Huxley model, there is typically a short interval in which a stable equilibrium and two stable limit cycles coexist. In our case, for the parameter values specified in Section 2.2, this happens between values $I = 7.8588$ and 7.933. These limit cycles are shown in Figure 3.7A.

The smallest of the two stable limit cycles in the picture only exists for a short I-interval, and has a very small attraction domain. Therefore, it is extremely hard to find by e.g. time integration. This implies that it is also not trivial to compute the PRC corresponding to that particular limit cycle. Our method however, has no problem computing it. In Figure 3.7B, the PRCs are depicted that correspond to the limit cycles from Figure 3.7A. The shapes of the two PRCs are very different. Also note that the darker PRC was actually larger in amplitude, but was rescaled to the same ranges as the gray PRC.

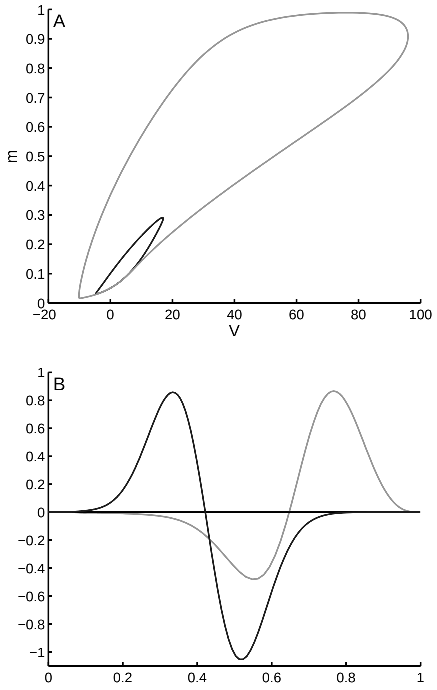


Figure 3.7: A) two stable limit cycles for the Hodgkin-Huxley model at parameter value $I = 7.919$. B) corresponding PRCs. The gray PRC corresponds to the gray limit cycle and the black PRC to the black limit cycle.

3.3 Continuation of homoclinic orbits

3.3.1 Homoclinic orbits

The general dynamical systems theory on homoclinic orbits is briefly recalled in Section 1.4.

There are two types of homoclinic orbits with codimension 1, namely homoclinic-to-hyperbolic-saddle (HHS), if x_0 is a saddle, and homoclinic-to-saddle-node (HSN), if x_0 is a saddle-node. Both these types of homoclinic orbits are important in many applications, e.g. as wave solutions in combustion models [6], to model ‘bursting’ in models of biological cells [110], chemical reactions [61], etc.

We have implemented continuation of both HHS and HSN orbits, starting from a Bogdanov-Takens (BT) point (no other software allows this) or from a limit cycle with high period, and the detection of a large number of codimension 2 bifurcations during the continuation. To compute the relevant eigenspaces of the equilibrium in each step, we use a method to continue invariant subspaces based on [29].

AUTO also has a toolbox for homoclinic continuation, namely HomCont [15]. Important differences with our implementation are that HomCont does not use the continuation of invariant subspaces, and cannot start the continuation of homoclinics from a BT-point. Also, we have implemented test functions for inclination-flip bifurcations in a new and more efficient way. Thus, the algorithm combines various ingredients from [29], [14], [15] and [9] but differs from any existing implementation.

3.3.2 Extended Defining System for Continuation

Homoclinic-to-Hyperbolic-Saddle Orbits

Defining system

To continue HHS orbits in two free parameters, we use an extended defining system that consists of several parts.

First, the infinite time interval is truncated, so that instead of $[-\infty, +\infty]$ we use $[-T, +T]$, which is rescaled to $[0, 1]$ and divided into mesh intervals. The mesh is non-uniform and adaptive. Each mesh interval is further subdivided by equidistant fine mesh points. Also, each mesh interval contains a number of collocation points. (This discretization is the same as that for limit cycles in Section 3.2.3.) The equation

$$\dot{x}(t) - 2Tf(x(t), \alpha) = 0, \tag{3.13}$$

must be satisfied in each collocation point.

The second part is the equilibrium condition

$$f(x_0, \alpha) = 0. \tag{3.14}$$

Third, there is a so-called phase condition for the homoclinic solution, similar to periodic solutions

$$\int_0^1 \dot{\widetilde{x}}^*(t)[x(t) - \widetilde{x}(t)]dt = 0. \tag{3.15}$$

Here $\widetilde{x}(t)$ is some initial guess for the solution, typically obtained from the previous continuation step. We note that in the literature another phase condition is also used, see, for example [39]. However, in the present implementation we employ the condition (3.15).

Fourth, there are the homoclinic-specific constraints to the solution. For these we need access to the stable and unstable eigenspaces of the system in the equilibrium point after each step. It is not efficient to recompute these spaces from scratch in each continuation step. Instead, we use the algorithm for continuing invariant subspaces, as described in [29].

Suppose in an n-dimensional system we have the following block Schur factorization for $A(0) = f_x(x_0, \alpha_0)$, the Jacobian at the equilibrium point of the known homoclinic orbit:

$$A(0) = Q(0)\; R(0)\; Q^T(0), \quad Q(0) = [Q_1(0)\; Q_2(0)] \tag{3.16}$$

where $A(0)$, $R(0)$ and $Q(0)$ are $n \times n$-matrices, $Q(0)$ is orthogonal, $Q_1(0)$ has dimensions $n \times m$ and $R(0)$ is block upper triangular

$$R(0) = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \tag{3.17}$$

where R_{11} is an $m \times m$-block (R_{11} and R_{22} are not required to be triangular). Then the columns of $Q_1(0)$ span an invariant subspace $P(0)$ of dimension m (e.g. the stable or unstable one) of $A(0)$, and the columns of $Q_2(0)$ span the orthogonal complement $P(0)^{\perp}$.

What we want for our next continuation step, are these subspace-defining columns for a matrix $A(s)$, close to $A(0)$ (due to the small stepsize in parameter space), without having to compute everything explicitly again; we will call them $Q_1(s)$ and $Q_2(s)$:

$$A(s) = Q(s)\, R(s)\, Q^T(s), \quad Q(s) = [Q_1(s)\; Q_2(s)]\;. \tag{3.18}$$

As described in [36], it is always possible to obtain a smooth path of block Schur factorizations, and we can accumulate all transformations in such a way that we are always looking for corrections close to the identity. Thus we can write (for s sufficiently small)

$$Q(s) = Q(0)\, U(s), \quad U(0) = I, \tag{3.19}$$

so that we now need to compute the $n \times n$-matrix $U(s)$. Partition $U(s)$ in blocks of the same size as we partitioned $R(0)$:

$$U(s) = [U_1(s)\; U_2(s)] = \begin{bmatrix} U_{11}(s) & U_{12}(s) \\ U_{21}(s) & U_{22}(s) \end{bmatrix} \tag{3.20}$$

so U_{11} has dimensions $m \times m$, and U_{22} dimensions $(n-m) \times (n-m)$.

We first prove that we can always assume that $U_{11}(s)$ and $U_{22}(s)$ are symmetric positive-definite (SPD), by redefining $Q(s)$ and $R(s)$ if necessary. The proof is as follows:

The polar decomposition of matrices [54] states that any matrix X can be written in a unique way as the product of a symmetric positive definite matrix P and an orthogonal matrix Z:

$$X = P\, Z\;.$$

Applying this to U_{11} and U_{22}, we can say

$$\begin{aligned} U_{11} &= V_{11}\,T_1\,, && (3.21)\\ U_{22} &= V_{22}\,T_2\,, && (3.22)\\ \Rightarrow \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} &= \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}, && (3.23) \end{aligned}$$

if we define $V_{12} = U_{12}\,T_2^T$ and $V_{21} = U_{21}\,T_1^T$.

$$\Rightarrow U(s) = V(s)\,T(s)\,, \tag{3.24}$$

where $T(s)$ is orthogonal, and also block-diagonal. Since $U(0) = I$, we can also claim $T(0) = I$.

Because of these properties, we can state that $Q'(s) = Q(s)T(s)$ is orthogonal, $R'(s) = T^T(s)R(s)T(s)$ is block upper triangular,

$$A(s) = Q'(s)\,R'(s)\,Q'^T(s)\,, \tag{3.25}$$

and $R'(s)$ has symmetric positive-definite upper left and lower right blocks.

Since $U(0) = I$, there is an interval I_0 around 0 where we can require for all $s \in I_0$ that

$$U_1(s) = \begin{bmatrix} I \\ U_{21}(s)U_{11}^{-1}(s) \end{bmatrix} U_{11}(s)\,. \tag{3.26}$$

Now define for all $s \in I_0$ the $(n-m) \times m$-matrix Y

$$Y = U_{21}(s)U_{11}^{-1}(s)\,. \tag{3.27}$$

Note that $U_1(s)$ is orthogonal, so:

$$U_1^T(s)U_1(s) = U_{11}^T(s)U_{11}(s) + U_{21}^T(s)U_{21}(s) = I\,. \tag{3.28}$$

Using (3.28), we obtain

$$\begin{aligned} I + Y^TY &= I + U_{11}^{-T}(s)U_{21}^T(s)U_{21}(s)U_{11}^{-1}(s) \\ &= I + U_{11}^{-T}(s)\,[I - U_{11}^T(s)U_{11}(s)]\,U_{11}^{-1}(s) \end{aligned}$$

$$\begin{aligned} &= I + U_{11}^{-T}(s)U_{11}^{-1}(s) - I \\ &= U_{11}^{-T}(s)U_{11}^{-1}(s) \ . \end{aligned}$$

Now because we have shown above that $U_{11}(s)$ can be chosen so that it is symmetric positive definite, $U_{11}^{-1}(s)$ is the unique positive definite square root of $I+Y^TY$. This implies that we can rewrite (3.26) in terms of Y

$$U_1(s) = \begin{bmatrix} I \\ Y \end{bmatrix} (I + Y^TY)^{(-1/2)} \ . \tag{3.29}$$

Now note that $U_2(s)$ is orthogonal, so:

$$U_2^T(s)U_2(s) = U_{12}^T(s)U_{12}(s) + U_{22}^T(s)U_{22}(s) = I \tag{3.30}$$

and note that $U_1^T(s)\, U_2(s) = 0$, so:

$$\begin{aligned} U_{11}^T(s)U_{12}(s) + U_{21}^T(s)U_{22}(s) &= 0 && (3.31) \\ U_{12}(s)U_{22}^{-1}(s) &= -U_{11}^{-T}(s)U_{21}^T(s) && (3.32) \\ U_{12}(s)U_{22}^{-1}(s) &= -(U_{21}(s)U_{11}^{-1}(s))^T = -Y^T \ . && (3.33) \end{aligned}$$

Since $U(0) = I$, there is an interval I_0 around 0 where we can require for all $s \in I_0$ that

$$U_2(s) = \begin{bmatrix} U_{12}(s)U_{22}^{-1}(s) \\ I \end{bmatrix} U_{22}(s) \ . \tag{3.34}$$

So from (3.33)

$$U_2(s) = \begin{bmatrix} -Y^T \\ I \end{bmatrix} U_{22}(s) \ . \tag{3.35}$$

Now, using (3.32) and (3.30), we can derive

$$\begin{aligned} I + Y\, Y^T &= I + U_{21}(s)U_{11}^{-1}(s)U_{11}^{-T}(s)U_{21}^T(s) \\ &= I + U_{22}^{-T}(s)U_{12}^T(s)U_{12}(s)U_{22}^{-1}(s) \\ &= I + U_{22}^{-T}(s)\, [I - U_{22}^T(s)U_{22}(s)]\, U_{22}^{-1}(s) \\ &= I + U_{22}^{-T}(s)U_{22}^{-1}(s) - I \\ &= U_{22}^{-T}(s)U_{22}^{-1}(s) \ . \end{aligned}$$

So if we choose $U_{22}(s)$ symmetric positive definite (as we have shown above to be possible), then $U_{22}^{-1}(s)$ is the unique positive definite square root of $I + Y\, Y^T$. This implies that we can rewrite (3.35) as

$$U_2(s) = \left[\begin{array}{c} -Y^T \\ I \end{array} \right] (I + Y\, Y^T)^{(-1/2)} \, . \tag{3.36}$$

Combining (3.29) and (3.36), we can write $U(s)$ completely in terms of Y:

$$U(s) = \left[\left(\begin{array}{c} I \\ Y \end{array} \right) (I + Y^T Y)^{(-1/2)} \quad \left(\begin{array}{c} -Y^T \\ I \end{array} \right) (I + YY^T)^{(-1/2)} \right] . \tag{3.37}$$

So to do a quick and smooth subspace continuation of both stable and unstable subspaces, we only need to keep track of two small matrices Y_S and Y_U (one per subspace) of dimension $(n - m) \times m$, which are smaller than the original subspaces. From these matrices Y, we can easily compute the span of stable and unstable subspaces, and their orthogonal complements.

We define $E(s)$ and $T(s)$ by

$$Q(0)^T\, A(s)\, Q(0) = R(0) + E(s) = \left[\begin{array}{cc} T_{11} & T_{12} \\ T_{21} & T_{22} \end{array} \right] . \tag{3.38}$$

Now if we regard the invariant subspace relation

$$Q_2^T(s)\, A(s)\, Q_1(s) = 0, \tag{3.39}$$

and we do substitutions using (3.19), (3.37) and (3.38), we obtain the following algebraic Riccati equation for Y:

$$T_{22}(s)\, Y - Y\, T_{11}(s) = -T_{21}(s) + Y\, T_{12}(s)\, Y \, . \tag{3.40}$$

Here T_{11} is of size $m \times m$ and T_{22} is an $(n - m) \times (n - m)$-matrix.

(3.40) is the equation added to the defining system to keep track of matrices Y for the stable and unstable eigenspaces: Y_S and Y_U are obtained from the separate Ricatti equations

$$\begin{array}{ccccccccccc} T_{22U} Y_U & - & Y_U T_{11U} & + & T_{21U} & - & Y_U T_{12U} Y_U & = & 0, \\ T_{22S}\, Y_S & - & Y_S\, T_{11S} & + & T_{21S} & - & Y_S\, T_{12S}\, Y_S & = & 0. \end{array} \tag{3.41}$$

Now that we have these matrices Y, we can formulate constraints on the behaviour of the solution close to the equilibrium x_0. The initial vector of the orbit, $(x(0) - x_0)$, is placed in the unstable eigenspace of the system in the equilibrium. We express that by the requirement that it is orthogonal to the orthogonal complement of the unstable eigenspace.

From(3.19) and (3.37), if $Q_U(0)$ is the orthogonal matrix from the previous step, related to the unstable invariant subspace, then a basis for the orthogonal complement of that subspace in the new step $Q_{U^\perp}(s)$, using Y_U, can be computed by

$$Q_{U^\perp}(s) = Q_U(0) \left[\begin{array}{c} -Y_U^T \\ I \end{array} \right].$$

Note that $Q_{U^\perp}(s)$ is not orthogonal. The full orthogonal matrix needed for the next step, is computed separately after each step. The matrices for the stable subspace can be computed similarly. The equations to be added to the system are

$$\begin{array}{l} Q_{U^\perp}(s)^T(x(0) - x_0) = 0, \\ Q_{S^\perp}(s)^T(x(1) - x_0) = 0. \end{array} \tag{3.42}$$

Finally, the distances between $x(0)$ (resp., $x(1)$) and x_0 must be small enough, so that

$$\begin{array}{l} \|x(0) - x_0\| - \epsilon_0 = 0, \\ \|x(1) - x_0\| - \epsilon_1 = 0. \end{array} \tag{3.43}$$

Implementation

The variables in this system are stored in one vector. It contains the values of $x(t)$ in the fine mesh points including $x(0)$ and $x(1)$, the truncation time T, two free system parameters, the coordinates of the saddle x_0, and the elements of the matrices Y_S and Y_U.

A system consisting of all equations (3.13), (3.14), (3.15), (3.41), (3.42) and (3.43), is overdetermined. The basic defining system for the continuation of a HHS orbit in two free system parameters consists of (3.13), (3.14), (3.41), (3.42), and (3.43) with ϵ_0 and ϵ_1 fixed, so that the phase condition (3.15) is not used.

Alternatively, the phase condition (3.15) is added automatically if from the triple (T, ϵ_0, ϵ_1) two are chosen to be variable, instead of just one. Any combination of one or two parameters of that triple is possible.

Homoclinic-to-Saddle-Node Orbits

For a homoclinic orbit to a saddle-node equilibrium, the extended defining system undergoes some small changes. Now $(x(0) - x_0)$ has to be placed in the center-unstable subspace, instead of the unstable space. Analogously, $(x(1)-x_0)$ must be in the center-stable subspace. Thus, one of these spaces has one extra eigenvector and one less restriction.

The vector-condition again is implemented by requiring that the vector is orthogonal to the orthogonal complement of the corresponding space. So the equations (3.42) themselves do not really change; the changes happen in the computation of the matrices Q. One of these matrices now has one less column, so there is one condition less.

The number of equations is restored however, by adding the constraint that the equilibrium must be a saddle-node. For this we use the bordering technique, as described in Section 4.2.1 of [56]. The technique basically requires g to be zero, where g is obtained by solving

$$\begin{pmatrix} f_x(x,\alpha) & w_{bor} \\ v_{bor}^T & 0 \end{pmatrix} \begin{pmatrix} v \\ g \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} . \tag{3.44}$$

Here w_{bor} and v_{bor} are bordering vectors, chosen in such a way that the matrix in (3.44) is nonsingular.

3.3.3 Starting Strategies

At present, continuation of homoclinic orbits in MATCONT can be started in two ways: either from a Bogdanov-Takens (BT) point or from a limit cycle with large period. A third option, namely using the homotopy method, starting from an unstable equilibrium, is under development (see Chapter 7 of this thesis).

Limit cycle with large period

When starting from a limit cycle with large period, the user must first declare the cycle to be a homoclinic-to-saddle orbit. Automatically, initial values for the homoclinic parameters will be computed. First the program looks for the point on the cycle with smallest $\|f(x,\alpha)\|$. This point is taken as a first approximation for the equilibrium x_0.

The mesh points of the limit cycle are kept as mesh points for the homoclinic orbit, except for the mesh interval that contains the current equilibrium approximation. This mesh interval will be omitted, as it will grow to infinity in the homoclinic orbit. In memory, the stored cycle then needs to be ’rotated’, so that the first and last point of the homoclinic orbit ($x(0)$ and $x(1)$) are effectively stored as first and last point, respectively. Half of the period of the remaining part of the cycle will be kept as initial value for T. Initial values for ϵ_0 and ϵ_1 are also computed; these can be found by simply computing the distance from $x(0)$ and $x(1)$ to the approximated equilibrium.

Then the user has to select 2 free system parameters, and 1 or 2 of the homoclinic parameters (T, ϵ_0 and ϵ_1). The size of the system (i.e. the number of equations) will automatically be adjusted according to the choice of the user. An example ’Starter’ window is shown in the left of Figure 3.8.

Bogdanov-Takens point

To start from a Bogdanov-Takens point, we use the method taken from [8]. It computes a predictor for the homoclinic orbit, using the coefficients of the normal form at the Bogdanov-Takens point. The predictor is defined as

$$\alpha = -\frac{5}{7}\epsilon^2 K_{1,1} + \frac{\epsilon^4}{49}\left(-6K_{1,0} + \frac{25}{2}K_2\right) + O(\epsilon^6) \tag{3.45}$$

$$\begin{aligned} x(t) &= \epsilon^2\left[-\frac{5}{7}H_{01,1} + \frac{1}{4a}\left(\xi_0\left(\frac{\epsilon}{2}t\right) + \frac{10}{7} + O(\epsilon)\right)q_0\right] \\ &\quad + \frac{\epsilon^3}{8a}\eta_0\left(\frac{\epsilon}{2}t\right)q_1 + O(\epsilon^4) \end{aligned} \tag{3.46}$$

where α is the deviation of the free parameter, and t is defined over $[0, 1]$.

The parameters in the above equations are obtained as follows:

ϵ determines the initial amplitude of the orbit, and is chosen by the user.
ξ_0 and η_0 are defined as

$$(\xi_0, \eta_0)(t) = 2\left(1 - \frac{3}{\cosh^2(t)}, 6\frac{\tanh(t)}{\cosh^2(t)}\right) \ . \tag{3.47}$$

q_0 and q_1 are real, linearly independent eigenvectors of the Jacobian matrix $A = f_x(x_0, \alpha_0)$ at the Bogdanov-Takens point, such that

$$\begin{aligned} A\,q_0 &= 0 \\ A\,q_1 &= q_0 \end{aligned}$$

and there are corresponding vectors p_0 and p_1 of the transposed matrix A^*

$$\begin{aligned} A^*\,p_1 &= 0 \\ A^*\,p_0 &= p_1 \ . \end{aligned}$$

We choose the vectors so that they satisfy

$$\begin{aligned} < q_0, p_0 > &= < q_1, p_1 >= 0 \\ < q_1, p_0 > &= < q_0, p_1 >= 1 \ . \end{aligned}$$

a can be computed as

$$a = \frac{1}{2} < p_1, B(q_0, q_0) > \tag{3.48}$$

where $B(q_0, q_0) = f_{xx}(x_0, \alpha_0)\ q_0\ q_0$, with $f_{xx}(x_0, \alpha_0)$ the Hessian matrix at the Bogdanov-Takens point.

$K_{1,1}$ and $K_{1,0}$ are columns from K_1:

$$K_1 = [K_{1,0}\ K_{1,1}] = (\gamma_1^2 + \gamma_2^2)^{-1} \begin{pmatrix} \gamma_1 & -\gamma_2 \\ \gamma_2 & \gamma_1 \end{pmatrix} \ , \tag{3.49}$$

where $(\gamma_1, \gamma_2) = p_1^* A_1$, and $A_1 = f_\alpha(x_0, \alpha_0)$.

$H_{01,1}$ comes from the matrix H_{01}:

$$H_{01} = [H_{01,0}\ H_{01,1}] = A^{-1}([q_1\ 0] - A_1 K_1)\ . \tag{3.50}$$

And finally K_2 is defined as

$$K_2 = -(p_1^* z) K_{1,0} \tag{3.51}$$

where

$$z = B(H_{01,1}, H_{01,1}) + 2B_1(H_{01,1}, K_{1,1}) + B_2(K_{1,1}, K_{1,1}) \tag{3.52}$$

and where $B = f_{xx}(x_0, \alpha_0)$, $B_1 = f_{x\alpha}(x_0, \alpha_0)$ and $B_2 = f_{\alpha\alpha}(x_0, \alpha_0)$.

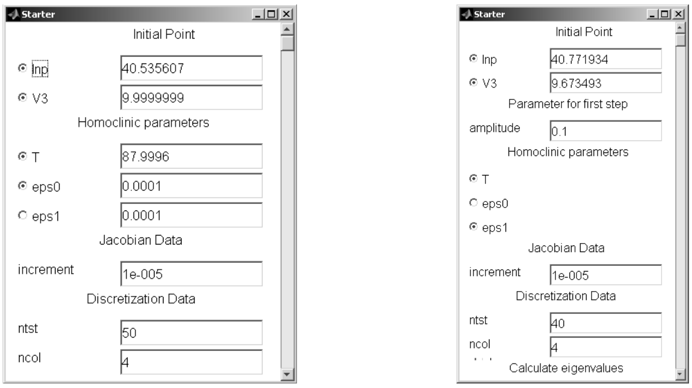
So by using all of these computations to obtain a predictor, the user only needs to define a value for the initial amplitude. This is the amplitude of the first computed homoclinic orbit, usually this shouldn't be too small (e.g. 0.1 or 0.01). Of the homoclinic parameters, by default T and ϵ_1 are the free parameters, but the user can choose any of the three parameters from the Starter window. An example 'Starter' window is shown in the right of Figure 3.8.

3.3.4 Bifurcations

Relationships between homoclinic objects of codimension 1 and 2 computed by MatCont 2.* are presented in Figure 3.9, while the symbols and their meaning are summarized in Table 3.3.

The arrows in Figure 3.9 have similar meaning as in Figure 3.1. In general, an arrow from an object of type A to an object of type B means that that object of type B can be detected (either automatically or by inspecting the output) during the computation of a curve of objects of type A. * stands for either S or U, depending on whether a stable or an unstable invariant manifold is involved.

In principle, the graphs presented in Figures 3.1 and 3.9 are connected. Indeed, it is known (see for example [2], [12]) that curves of codim 1 homoclinic bifurcations emanate from the BT, ZH, and HH codim 2 points. The current version of MatCont fully supports, however, only one such connection: BT →



Starter

Initial Point

● Inp 40.535607

● V3 9.9999999

Homoclinic parameters

● T 87.9996

● eps0 0.0001

○ eps1 0.0001

Jacobian Data

increment 1e-005

Discretization Data

ntst 50

ncol 4

Starter

Initial Point

● Inp 40.771934

● V3 9.673493

Parameter for first step

amplitude 0.1

Homoclinic parameters

● T

○ eps0

● eps1

Jacobian Data

increment 1e-005

Discretization Data

ntst 40

ncol 4

Calculate eigenvalues

Figure 3.8: (Parts of) the starter windows when continuing a curve of HHS orbits, starting from a limit cycle or homoclinic orbit (left) or from a BT point (right).

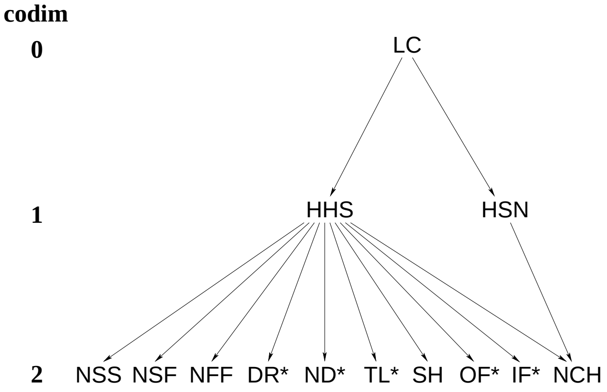


Figure 3.9: The graph of adjacency for homoclinic bifurcations in MatCont; here * stands for S or U.

Type of object	Label
Limit cycle	LC
Homoclinic to Hyperbolic Saddle	HHS
Homoclinic to Saddle-Node	HSN
Neutral saddle	NSS
Neutral saddle-focus	NSF
Neutral Bi-Focus	NFF
Shilnikov-Hopf	SH
Double Real Stable leading eigenvalue	DRS
Double Real Unstable leading eigenvalue	DRU
Neutrally-Divergent saddle-focus (Stable)	NDS
Neutrally-Divergent saddle-focus (Unstable)	NDU
Three Leading eigenvalues (Stable)	TLS
Three Leading eigenvalues (Unstable)	TLU
Orbit-Flip with respect to the Stable manifold	OFS
Orbit-Flip with respect to the Unstable manifold	OFU
Inclination-Flip with respect to the Stable manifold	IFS
Inclination-Flip with respect to the Unstable manifold	IFU
Non-Central Homoclinic to saddle-node	NCH

Table 3.3: Objects related to homoclinics to equilibria and their labels within the GUI

HHS.

During HSN continuation, only one bifurcation is tested for, namely the non-central homoclinic-to-saddle-node orbit or NCH. This orbit forms the transition between HHS and HSN curves. The strategy used for detection is taken from HomCont [15].

During HHS continuation, all bifurcations detected in HomCont are also detected in our implementation. For this, mostly test functions from [15] are used.

Suppose that the eigenvalues of $f_x(x_0, \alpha_0)$ can be ordered according to

$$\Re(\mu_{ns}) \leq ... \leq \Re(\mu_1) < 0 < \Re(\lambda_1) \leq ... \leq \Re(\lambda_{nu}), \tag{3.53}$$

where $\Re()$ stands for 'real part of', ns is the number of stable, and nu the number of unstable eigenvalues. The test functions for the bifurcations are

- Neutral saddle, saddle-focus or bi-focus

  $$\psi = \Re(\mu_1) + \Re(\lambda_1)$$

  If both μ_1 and λ_1 are real, then it is a neutral saddle, if one is real and one consists of a pair of complex conjugates, the bifurcation is a saddle-focus, and it is a bi-focus when both eigenvalues consist of a pair of complex conjugates.

- Double real stable leading eigenvalue

  $$\psi = \begin{cases} (\Re(\mu_1) - \Re(\mu_2))^2, & \Im(\mu_1) = 0 \\ -(\Im(\mu_1) - \Im(\mu_2))^2, & \Im(\mu_1) \neq 0 \end{cases}$$

- Double real unstable leading eigenvalue

  $$\psi = \begin{cases} (\Re(\lambda_1) - \Re(\lambda_2))^2, & \Im(\lambda_1) = 0 \\ -(\Im(\lambda_1) - \Im(\lambda_2))^2, & \Im(\lambda_1) \neq 0 \end{cases}$$

- Neutrally-divergent saddle-focus (stable)

  $$\psi = \Re(\mu_1) + \Re(\mu_2) + \Re(\lambda_1)$$

- Neutrally-divergent saddle-focus (unstable)

  $$\psi = \Re(\mu_1) + \Re(\lambda_2) + \Re(\lambda_1)$$

- Three leading eigenvalues (stable)

  $$\psi = \Re(\mu_1) - \Re(\mu_3)$$

- Three leading eigenvalues (unstable)

  $$\psi = \Re(\lambda_1) + \Re(\lambda_3)$$

- Non-central homoclinic-to-saddle-node

$$\psi = \Re(\mu_1)$$

- Shil'nikov-Hopf

$$\psi = \Re(\lambda_1)$$

- Bogdanov-Takens point

$$\psi = \begin{cases} \Re(\mu_1) \\ \Re(\lambda_1) \end{cases}$$

For orbit- and inclination-flip bifurcations, we assume the same ordering of the eigenvalues of $f_x(x_0, \alpha_0) = A(x_0, \alpha_0)$ as shown in (3.53), but also that the leading eigenvalues μ_1 and λ_1 are unique and real:

$$\Re(\mu_{ns}) \leq ... \leq \Re(\mu_2) < \mu_1 < 0 < \lambda_1 < \Re(\lambda_2) \leq ... \leq \Re(\lambda_{nu}) \; .$$

Then it is possible to choose normalised eigenvectors p_1^s and p_1^u of $A^T(x_0, \alpha_0)$ and q_1^s and q_1^u of $A(x_0, \alpha_0)$ depending smoothly on (x_0, α_0), which satisfy

$$\begin{array}{rr} A^T(x_0, \alpha_0)\, p_1^s = \mu_1\, p_1^s & \quad A^T(x_0, \alpha_0)\, p_1^u = \lambda_1\, p_1^u \\ A(x_0, \alpha_0)\, q_1^s = \mu_1\, q_1^s & \quad A(x_0, \alpha_0)\, q_1^u = \lambda_1\, q_1^u \; . \end{array}$$

The test functions for the orbit-flip bifurcations are then:

- Orbit-flip with respect to the stable manifold

$$\psi = e^{-\mu_1 T} < p_1^s, x(1) - x_0 >$$

- Orbit-flip with respect to the unstable manifold

$$\psi = e^{\lambda_1 T} < p_1^u, x(0) - x_0 >$$

For the inclination-flip bifurcations, in [15] the following test functions are introduced:

- Inclination-flip with respect to the stable manifold

$$\psi = e^{-\mu_1 T} < q_1^s, \phi(0) >$$

- Inclination-flip with respect to the unstable manifold

$$\psi = e^{\lambda_1 T} < q_1^u, \phi(1) >$$

where ϕ ($\phi \in \mathcal{C}^1([0,1], \mathbb{R}^n)$) is the solution to the adjoint system, which can be written as

$$\left\{ \begin{array}{l} \dot{\phi}(t) + 2\,T\,A^T(x(t),\alpha_0)\,\phi(t) = 0 \\ (L_s)^T \phi(1) = 0 \\ (L_u)^T \phi(0) = 0 \\ \int_0^1 \widetilde{\phi}^T(t)[\phi(t) - \widetilde{\phi}(t)]dt = 0 \end{array} \right. \tag{3.54}$$

where L_s and L_u are matrices whose columns form bases for the stable and unstable eigenspaces of $A(x_0, \alpha_0)$, respectively, and the last condition selects one solution out of the family $c\phi(t)$ for $c \in \mathbb{R}$. L_u is equivalent to Q_U from Section 3.3.2 and L_s to Q_S. In the homoclinic defining system the orthogonal complements of Q_S and Q_U are used; in the adjoint system for the inclination-flip bifurcation, we use the matrices themselves (or at least, their transposed versions).

However, it is possible to compute the above-mentioned test functions in a more efficient way, without having to explicitly find the solution ϕ to the adjoint system.

If ϕ is a solution to (3.54) and $\zeta_1 \in \mathbb{R}^{nu}$, $\zeta_2 \in \mathbb{R}^{ns}$, then

$$\left(\begin{array}{c} \phi(t) \\ \zeta_1 \\ \zeta_2 \end{array} \right) \perp Range \left(\begin{array}{c} D - 2\,T\,A(x(t),\alpha_0) \\ (L_s^\perp)^T\,\delta_1 \\ (L_u^\perp)^T\,\delta_0 \end{array} \right) \tag{3.55}$$

$$\Longleftrightarrow$$

$$\left\{ \begin{array}{l} L_s^\perp \zeta_1 = -\phi(1) \\ L_u^\perp \zeta_2 = \phi(0) \end{array} \right. \tag{3.56}$$

Here D and δ are the differentiation and the evaluation operators, respectively, and $L_s^\perp$ and $L_u^\perp$ are the orthogonal complements of L_s and L_u, respectively.

Proof

The necessary and sufficient condition is that for all $g \in \mathbb{C}^1([0,1], \mathbb{R}^n)$

$$\left(\begin{array}{c} \phi(t) \\ \zeta_1 \\ \zeta_2 \end{array} \right) \perp \left(\begin{array}{c} g'(t) - 2\,T\,A(x(t), \alpha_0) g(t) \\ (L_s^{\perp})^T\, g(1) \\ (L_u^{\perp})^T\, g(0) \end{array} \right), \tag{3.57}$$

i.e.

$$\int_0^1 \phi(t)^T (g'(t) - 2\,T\,A(x(t), \alpha_0) g(t)) dt + \zeta_1^T\, (L_s^{\perp})^T\, g(1) + \zeta_2^T\, (L_u^{\perp})^T\, g(0) = 0 \tag{3.58}$$

$$\begin{aligned} \left[\phi(t)^T g(t)\right]_0^1 - \int_0^1 (\phi'(t)^T g(t) + \phi(t)^T 2\,T\,A(x(t), \alpha_0) g(t)) dt \\ + \zeta_1^T\, (L_s^{\perp})^T\, g(1) + \zeta_2^T\, (L_u^{\perp})^T\, g(0) = 0 \end{aligned} \tag{3.59}$$

where

$$\begin{array}{rl} \phi'(t)^T g(t) + \phi(t)^T\, 2\,T & A(x(t), \alpha_0) g(t) \\ = & g(t)^T \left(\phi'(t) + 2\,T\,A^T(x(t), \alpha_0)\phi(t)\right) \\ = & 0\,. \end{array} \tag{3.60}$$

So the condition is

$$\phi(1)^T g(1) - \phi(0)^T g(0) + \zeta_1^T\, (L_s^{\perp})^T\, g(1) + \zeta_2^T\, (L_u^{\perp})^T\, g(0) = 0 \tag{3.61}$$

$$(\phi(1)^T + \zeta_1^T\, (L_s^{\perp})^T) g(1) - (\phi(0)^T + \zeta_2^T\, (L_u^{\perp})^T) g(0) = 0\,. \tag{3.62}$$

This must hold for all g, hence

$$\left\{ \begin{array}{l} \zeta_1^T\, (L_s^{\perp})^T = -\phi(1)^T \\ \zeta_2^T\, (L_u^{\perp})^T = \phi(0)^T \end{array} \right. \tag{3.63}$$

or

$$\left\{ \begin{array}{l} L_s^{\perp} \zeta_1 = -\phi(1) \\ L_u^{\perp} \zeta_2 = \phi(0) \end{array} \right. \tag{3.64}$$

□

We note that by (3.54) $\phi(1)$ and $\phi(0)$ are vectors, orthogonal to the stable and unstable eigenspaces, respectively, and so (3.64) is solvable. In fact, the solutions are unique since $L_s^\perp$ and $L_u^\perp$ are full rank matrices.

Thus ζ_1 and ζ_2 are determined uniquely, so that the test functions for inclination-flip can be rewritten as

- Inclination-flip with respect to the stable manifold

$$\psi = e^{-\mu_1 T} < q_1^s, \phi(0) >= e^{-\mu_1 T} < q_1^s, L_u^\perp \zeta_2 >$$

- Inclination-flip with respect to the unstable manifold

$$\psi = e^{\lambda_1 T} < q_1^u, \phi(1) >= -e^{\lambda_1 T} < q_1^u, L_s^\perp \zeta_1 >$$

These test functions are very efficiently computable: $L_u^\perp$ and $L_s^\perp$ are already computed during homoclinic continuation, for the continuation of the invariant subspaces (see Section 3.3.2); μ_1, λ_1, q_1^s and q_1^u are also already known. The only quantities that seem harder to compute are ζ_1 and ζ_2.

But the advantage of our computation becomes clear when comparing the matrix in (3.55) to the equations (3.13) and (3.42), which are part of the defining system for homoclinic orbits. Indeed, the big matrix in the discretization of (3.55) is for a major part identical to the Jacobian matrix used during homoclinic continuation, and therefore almost no extra time is required to compute it. After selecting the necessary rows and columns from the already computed Jacobian, the resulting matrix is bordered with its left and right singular vectors, which results in a non-singular matrix (call it B). Then solving

$$B^T \setminus [\,0 \ldots 0\ 1\,]^T$$

results in finding ζ_1 and ζ_2 as two- and one-but-last block elements of the resulting vector. This vector is also the new left singular vector of B, after eliminating the last element and normalizing the vector. Similarly, the new right singular vector can be found by solving $B \setminus [\,0 \ldots 0\ 1\,]^T$, removing the last element and renormalizing. Thus our algorithm, while computing ζ_1 and ζ_2 relatively cheaply, also keeps track of the singular vectors which facilitate getting a well-conditioned matrix B.

3.3.5 The Koper example

Consider the following system of differential equations:

$$\left\{ \begin{array}{rcl} \epsilon_1 \dot{x} & = & (k\, y - x^3 + 3x - \lambda) \\ \dot{y} & = & x - 2y + z \\ \dot{z} & = & \epsilon_2 (y - z). \end{array} \right. \tag{3.65}$$

This system, which is a three-dimensional van der Pol-Duffing oscillator, has been introduced and studied in [83]. It is used as a standard demo in HOMCONT. Parameters ϵ_1 and ϵ_2 are kept at 0.1 and 1, respectively. We note that system (3.65) has a certain symmetry: If $(x(t), y(t), z(t))$ is a solution for a given value of λ, then $(-x(t), -y(t), -z(t))$ is a solution for $-\lambda$.

Starting from a general point $(0, -1, 0.1)$ and setting $k = 0.15$ and $\lambda = 0$, we find by time integration a stable equilibrium at $x_0 = y_0 = z_0 = -\sqrt{3.15} \approx -1.77482$.

By equilibrium continuation with λ free, we find two limit points (LP), at $(-1.024695,\ -1.024695,\ -1.024695)$ for $\lambda = -2.151860$ with normal form coefficient $a = -4.437056e + 000$ and at $(1.024695,\ 1.024695,\ 1.024695)$ for $\lambda = 2.151860$ with normal form coefficient $a = +4.437060e + 000$ (note the reflection). The meaning of these normal form coefficients is explained in Section 1.2.

By continuation of the limit points with (k, λ) free, MATCONT detects a cusp point CP at $(0, 0, 0)$ for $k = -3$ and $\lambda = 0$. Also detected are two Zero-Hopf points ZH for $k = -0.3$ at $\pm(0.948683,\ 0.948683,\ 0.948683)$ and $\lambda = \pm 1.707630$, but these are in fact Neutral Saddles. Further, two Bogdanov-Takens points BT are found for $k = -0.05$ at $\pm(0.991632,\ 0.991632,\ 0.991632)$ and $\lambda = \pm 1.950209$. The normal form coefficients are $(a, b) = (6.870226e + 000, 3.572517e + 001)$.

A bifurcation diagram of (3.65) is shown in Figure 3.10.

We now compute a HHS curve starting at one of the BT points, and setting the curve-type to Homoclinic. To further initialize the homoclinic continuation, some parameters have to be set and/or selected in the 'Starter' window.

First we select 2 system parameters as free parameters, e.g. k and λ. We must

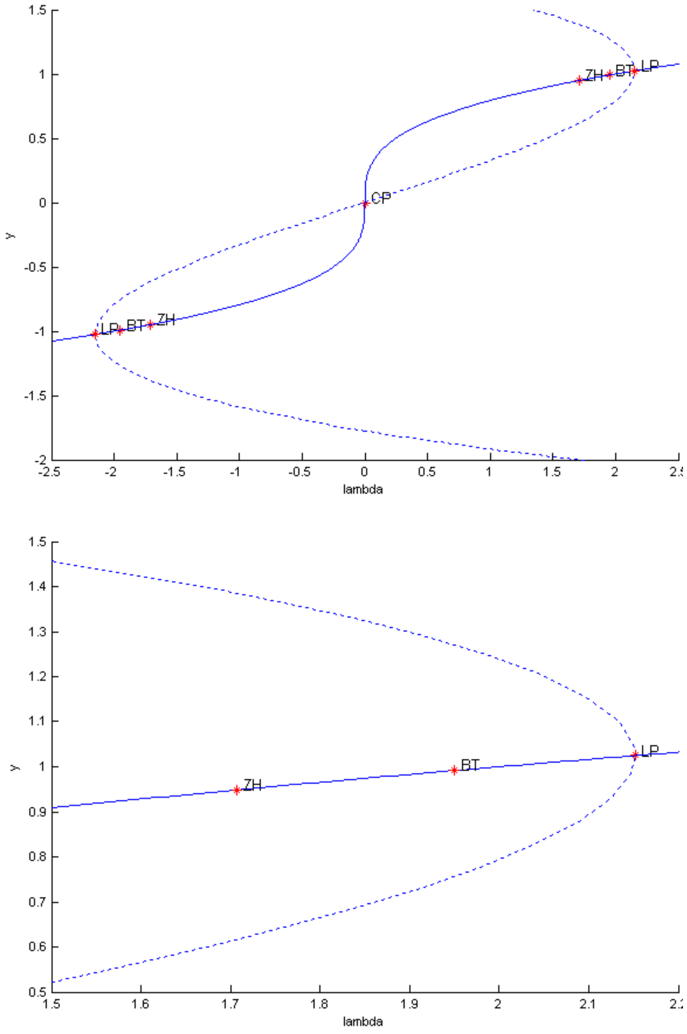


Figure 3.10: Top: equilibrium bifurcation diagram of the Koper system. Dashed line = equilibria, full line = limit points, CP = Cusp Point, BT = Bogdanov-Takens, ZH = Zero-Hopf, LP = Limit Point. Bottom: zoom on the top part of the diagram.

also set the *Initial amplitude*, i.e. an approximate size of the first homoclinic orbit. The default setting of 0.1 is a good value in most cases. In this case, the continuation is easiest to start by choosing ϵ_0 and ϵ_1 as free homoclinic parameters. Best convergence (and longest continuation) are reached when a few adjustments are made to settings in the continuer window. These recommended changes are to increase the tolerances (e.g. to 1e-3) and to increase the maximum stepsize (e.g. to 10).

Another option is to start from a limit cycle with large period. For example, select the BT point at $\lambda = 1.950209$, and then compute a curve of Hopf points H passing through it, along which one encounters a Generalized Hopf bifurcation GH. Stop the continuation at (the arbitraty value) $\lambda = 1.7720581$, where the Hopf point is at $(0.98526071, 0.98526071, 0.98526071)$ and $k = -0.23069361$. We then can find limit cycles for very slowly decreasing values of λ, (λ decreases down to 1.77178), with a rapid increase in the period. At some point, one can stop the continuation, and switch to the continuation of the homoclinic orbit, e.g. when the period of the limit cycle is higher than 100.

Important to note that the choice of free homoclinic parameters, especially at the start-up of a homoclinic continuation, is crucial. In this Koper example, the continuation of homoclinic orbits from this large periodic orbit will give the best result (i.e. find the largest set of consecutive points) if ϵ_0 and ϵ_1 are chosen as free homoclinic parameters. The reason for this is not completely clear yet, but it is significant that in this case the Jacobian has a condition number of a much lower order than in all other five possible scenarios (T and ϵ_0, T and ϵ_1, T, ϵ_0 or ϵ_1 as free parameters).

One can also monitor the eigenvalues of the equilibrium during continuation, by displaying them in the ’Numerical’ window. This is a very useful feature, because it gives indications on what further bifurcations might be expected. For example, a non-central homoclinic-to-saddle-node reveals itself by the fact that one eigenvalue approaches zero. Once one is close enough to such a point, the user can switch to the continuation of HSN orbits from there.

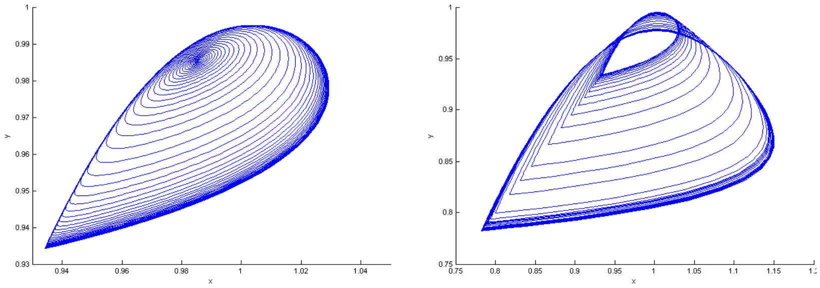


Figure 3.11: Left: limit cycles, starting from a Hopf point H and approaching a homoclinic orbit. Right: a family of HHS orbits.

3.3.6 The Morris-Lecar example

There are also several curves of homoclinic orbits present in the Morris-Lecar model (described in Section 2.2.3).

When starting at parameter values $I = 0$ and $V_3 = 6$, through time integration you can find a stable equilibrium at coordinates $(V, N) \approx (-59.49, 0.00054)$. From this point, freeing parameter I an equilibrium curve can be computed. For this curve, you will probably want to increase the default maximum stepsize in the continuer window, e.g. to 10. Along the curve, several equilibrium bifurcations are detected for increasing I: two limit points (at $I = 43.740592$ and $I = 34.546930$) and two Hopf points (at $I = 43.312018$ and $I = 197.796288$). A picture of this equilibrium curve is shown in Figure 3.12.

When you select the second Hopf point, a limit cycle curve can be started from there. Again, the maximum stepsize can be increased to allow a faster computation of the smooth curve. As free parameters, we keep I, and add the period of the cycle. The first limit point of cycles that is detected is actually the Hopf point the curve started from. First I increases, but after another LPC, the curve turns and I starts to decrease. The period of the limit cycles along the curve first decreases, but then starts to rise. When the default setting was kept, to compute 300 points, and the maximum stepsize was set at 10, the continuation will end after 300 points at an LC with period 135.19178, at parameter value $I = 42.936367$. The resulting limit cycles are shown in phase space in Figure 3.13.

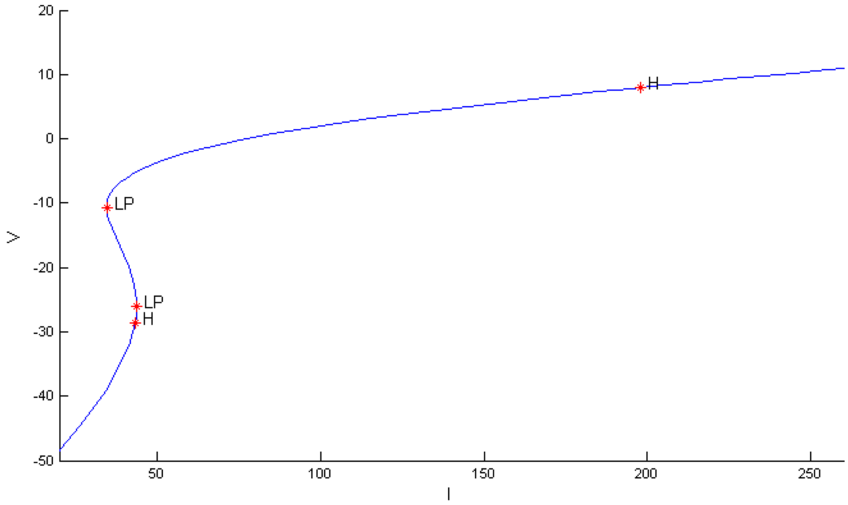


Figure 3.12: (Partial) equilibrium curve with bifurcations of the Morris-Lecar system in parameter space.

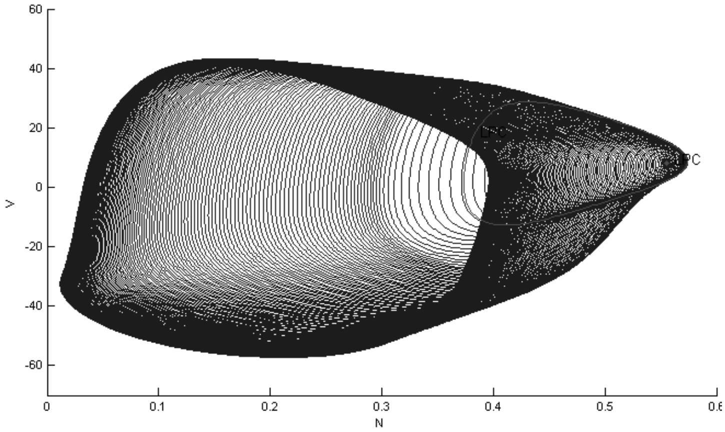


Figure 3.13: (Partial) limit cycle curve with bifurcations of the Morris-Lecar system in phase space.

This high period indicates the possible presence of a homoclinic orbit. So you can select the last limit cycle of the curve, and define it to be a homoclinic-to-saddle orbit. Then you have to set several parameters: choose both I and V_3 as free parameters, and choose ϵ_0 and ϵ_1 as free homoclinic parameters. In the continuer window, you need to increase the tolerances, to e.g. 0.001. You should also make sure the number of mesh intervals is not too small (e.g. 40). But again the maximum stepsize can be increased from the default value (to e.g. 1). This way a curve of homoclinic orbits can be computed.

This system seems to be less sensitive than the previous example, the Koper system. A consequence is that in the initiation of the homoclinic continuation from the large periodic orbit, the user has several possible choices for free homoclinic parameters, which will all allow a clean run of the continuation. Possible choices are ϵ_0 and ϵ_1, T and ϵ_0 or ϵ_1 alone. In the other three possibilities, the continuation will be harder to start.

You can also, going back to the equilibrium curve, select the first Hopf point. From that point you can start the continuation of a curve of Hopf points. Along that curve 4 different Bogdanov-Takens points can be detected, at the I-values 512.545668, 497.365842, -204.645815 and 40.771934. When you select the last one, at $I = 40.771934$, you can choose to compute a curve of homoclinic-to-saddle orbits. You can choose T and ϵ_1 as free homoclinic parameters, set the initial amplitude to 0.1 and the tolerances to 0.001. Then the continuation starts nicely.

If a continuation breaks down with the message `stepsize too small`, you can select the last point on the computed curve, and restart, e.g. with different free homoclinic parameters or tolerance settings.

3.4 Speed improvement by C-code

3.4.1 Incorporating C-code in MATLAB

One disadvantage of MATCONT is the lack of speed. This is mainly due to the fact that the MATLAB programming language is an interpreted language. This type of language will, in execution time, always be slower than compiled programs.

So to speed up the program, we need to step away from the idea of writing everything in MATLAB. The basic version of MATLAB contains its own C-compiler. It is possible, through some special intermediate kind of C-code, to have the C-code and MATLAB-code communicating with each other. Because the compiler is present in every version of MATLAB, we can supply the C-code and let the program compile the C-code at start-up. This causes a slight loss in time, negligible when compared to the time needed to continue limit cycles. The advantage of this approach is that it keeps the program completely platform-independent. The compiled C-code, which is platform-dependent, is only generated at runtime, and is thus adapted to the machine.

The compiled function can be called from the MATLAB-code, as if it were a MATLAB-function. We note that C-code written to communicate with a MATLAB-program, is not pure C-code. A standard C-compiler would object to parts of that code because of some intermediate classes of variables, and intermediate types of commands, which are not defined in standard C. There is for example the type `MxArray`, which is the standard data type to pass parameter lists between a C-function and MATLAB. Memory allocation happens through special commands such as `MxCreateDoubleMatrix` and `MxCreateSparse`.

The key lies in the parameter lists. The C-code receives all parameters from MATLAB in one big object of the type pointer to `MxArray`, and also passes all return variables back to the calling MATLAB-code in such an object. The difficulty when writing the special C-code is to analyze that object, to decide what kinds of parameters are actually passed, and to store them in the correct (types of) C-variables before doing any real computations in C. Analogously, before closing the C-function, all return-variables must be stored in a pointer to `MxArray`, in such a way that the calling MATLAB-function will be able to read and recognize all variables.

In the C-code, it is sometimes necessary or just better to call a MATLAB-function or operator. For example, to solve a big linear system it can be faster to use the backslash operator in MATLAB than to use self-written C-code. And again, there is a solution. As it is possible to call compiled C-functions from MATLAB, it is also possible to call MATLAB operators or self-written MATLAB-functions from

C, as long as the C-code makes sure that all parameters are stored properly, before passing them on to the MATLAB-code, and that the C-code processes the return values correctly.

3.4.2 C-code in MATCONT

In the pure MATLAB version of CL_MATCONT, over half the time spent in the continuation of limit cycles is actually spent in the evaluation of the Jacobian matrix of the discretized boundary-value problem, implemented in `BVP_LC_jac.m`. Thus, this function is a natural choice for re-implementation in C. Other matrix evaluations implemented in C in recent versions of MATCONT are similar to `BVP_LC_jac.m`:

- `BVP_PD_jac.m`: used in the continuation of PD cycles,
- `BVP_BPC_jacC.m`: used in the location of BPC cycles,
- `BVP_BPC_jacCC.m`: used in the continuation of BPC cycles,
- `BVP_LPC_jac.m`: used in the continuation of LPC cycles,
- `BVP_NS_jac.m`: user in the continuation of NS cycles.

This results in a significant gain in efficiency. In Tables 3.4 and 3.5, it is shown that the relative improvement by using the C-code is around 25% when turning off bifurcation detection, and over 50% when the test functions are turned on.

In the same table, a timing comparison with the CONTENT-package [85] is shown under a comparable GUI-load. The *adapt* system is a very simple model from adaptive control, considered as a test example in [31]. The *Connor* system is a complicated neural model with six state variables, listed in Section 2.2.2. All computations were done under Windows XP on a Intel(R) Pentium (R)M computer with a 1.73GHz computer with clock speed 795 MHz.

It is clear that on level of timing a MATLAB-program, even when using the C-code, cannot compete with the fully compiled software CONTENT. However, by using MATLAB a lot can be gained in ease of installation, user-friendliness,

Mesh intervals	test functions	MATCONT 2.2.2 (C-code)	MATCONT 1.1 (M-code)	improvement	CONTENT
50	off	32 s	44 s	27%	6 s
50	on	53 s	121 s	56%	7 s
100	off	86 s	115 s	25%	11 s
100	on	145 s	340 s	57%	13 s
200	off	265 s	339 s	22%	14 s
200	on	464 s	1077 s	57%	20 s

Table 3.4: Timing results for limit cycle-continuation, for 200 limit cycles in the adapt-system. The first column states the number of mesh intervals. The second column mentions whether the test functions for bifurcations and the multipliers were turned on. The third, fourth and fifth column give timing results for the newest version of MATCONT, for the previous (full MATLAB) version of MATCONT, the relative improvement between versions, and the timing result for CONTENT.

extendibility and functionalities in general. Also conversion of data is an important issue, and many scientists already know MATLAB, which lowers the threshold to start using MATCONT. Moreover, the C-re-implementation of additional functions of MATCONT can be done independently, function per function, to further improve the performance of MATCONT without disabling any of its features.

3.5 Sparse matrix solvers in MATLAB

3.5.1 Sparse matrices in MATCONT

Periodic orbits consist of an infinite number of points. In MATCONT, a finite number of these is stored in memory and used for continuation. These points are defined by mesh and collocation points (see Section 3.2.3 for details).

Due to the discrete representation of the solution curves, the Jacobian of the system, used during continuation steps, becomes finite (but large) and sparse. The columns of this matrix represent the unknowns of the discretized problem (including all mesh points), the rows represent the equations of the extended system (including one row per evaluation in a collocation point). E.g. for 3 mesh intervals and 2 collocation points in a system with 2 variables, the Jacobian matrix has the sparsity structure presented in Figure 3.14, where •’s are a priori non-zeros. The

Mesh intervals	test functions	MATCONT 2.2.2 (C-code)	MATCONT 1.1 (M-code)	improve- ment	CONTENT
50	off	45 s	55 s	18%	7 s
50	on	78 s	170 s	54%	11 s
100	off	133 s	169 s	21%	13 s
100	on	250 s	564 s	56%	22 s

Table 3.5: Timing results for limit cycle continuation, for 100 limit cycles in the Connor-system. The first column states the number of mesh intervals. The second column mentions whether the test functions for bifurcations and the multipliers were turned on. The third, fourth and fifth column give timing results for the newest version of MATCONT, for the previous (full MATLAB) version of MATCONT, the relative improvement between versions, and the timing result for CONTENT.

last row is a full row, added during the continuation steps.

In each continuation step, a system of the form $A\,x = b$ needs to be solved, where A is a large Jacobian matrix with the explained sparsity structure. This shows that the efficiency in solving such large sparse systems plays a crucial role in the overall performance of MATCONT for LC continuation experiments.

In Section 5.2.5 in [34], a study is done on different ways to exploit the structure and/or sparsity of this matrix in the system. Three solution methods were tested: the MATLAB regular (dense) solver, the MATLAB sparse solver, and a self-implemented method that makes use of the special structure of the sparse matrix. The result was that the self-implemented method, that in theory would result in the best performance, was actually the slowest. This is caused by the fact that the two other methods are built-in MATLAB functions, fully optimized and internally compiled. Between those two MATLAB methods, the sparse solver turned out to be much faster than the dense solver. Thus, in MATCONT the sparse solvers were used during continuation experiments.

3.5.2 Problem with sparse solvers

MATLAB has a number of built-in sparse solvers. One of the most recent additions is UMFPACK. UMFPACK is the sparse package that is used by the sparse LU factorization and sparse general solve $X = A \setminus B$ when A is sparse, square but

$$\left(\begin{array}{cccccccccccccccc}
x^0 & & x^{0,1} & & x^{1,0} & & x^{1,1} & & x^{2,0} & & x^{2,1} & & x^{3,0} & & T & \alpha \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & & & & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & & & & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & & & & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & & & & \bullet & \bullet \\
& & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & \bullet & \bullet \\
& & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & \bullet & \bullet \\
& & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & \bullet & \bullet \\
& & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & \bullet & \bullet \\
& & & & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & & & & & & & & & & & \bullet & \bullet & & \\
\bullet & \bullet & & & & & & & & & & & \bullet & \bullet & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{array}\right)$$

Figure 3.14: Jacobian of the discretized equations

not diagonal. Though it has become the standard sparse solver in MATLAB, we have found that this package does not always work properly.

Consider the three-dimensional Roessler system [117]:

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + \alpha y \\ \dot{z} = \beta + z(x - \gamma) \end{cases} \tag{3.66}$$

When starting from parameter values $\alpha = -1$, $\beta = 0.25$ and $\gamma = 6.2$, one quickly finds a stable equilibrium by doing time integration. This equilibrium can be continued for increasing α, locating a Hopf bifurcation at $\alpha \approx 0.00635$. From this Hopf point, limit cycle continuation can be started. These limit cycles are nicely circular in shape, when staying close to the Hopf point, but strangely enough MATLAB struggles with this continuation experiment when its settings are kept standard, and it breaks down after a small number of steps, around $\alpha = 0.01287$.

When one first disables the standard use of the UMFPACK (by the command `spparms('umfpack',0)`, so MATLAB will use other sparse solvers, the continuation goes much faster, better and exactly as one would expect. A resulting diagram is shown in Figure 3.15.

Upon closer inspection, we found that MATLAB, when trying to solve the large sparse system during the continuation, detects (falsely) that at some point during the continuation, the matrix becomes `close to singular or badly scaled.` When doing the sparse LU factorization in UMFPACK, MATLAB estimates that the condition number of the large sparse matrix would be approximately the inverse of $5.779657e - 017$, and thus huge. Therefore, MATLAB stops the computation of this step, and tries again with a smaller stepsize. Since it always gets this result, it stops computing when the stepsize gets below the allowed threshold.

However, by letting MATLAB compute the condition number of the large sparse matrix exactly, by regarding it as dense instead of sparse (which is not very efficient, but accurate), we found that the actual condition number of the matrix is around the order of $2.8354e + 003$, which shows that it is not by far as ill-conditioned as MATLAB seems to estimate, and continuation should work fine.

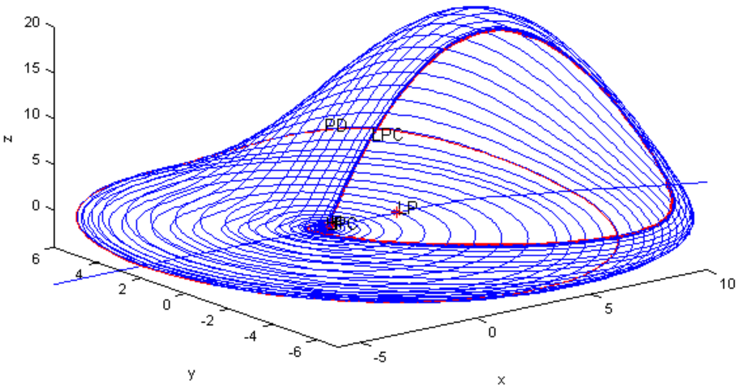


Figure 3.15: 3D diagram showing equilibrium curve and limit cycles for the Roessler system (3.66).

Apparently the general sparse solvers do work fine when you disable the UMFPACK, though in some cases UMFPACK does cause a speed-up of computations. So now at the start-up of MATCONT, the UMFPACK is automatically disabled, to avoid unnecessary breakdown of the continuation by the MATLAB sparse routines.

Chapter 4

Applications

A software package like MATCONT has potential applications in many different fields of science. Systems of ODEs have applications in economics, engineering, biology, chemistry, and physics. In the rapidly expanding field of mathematical biology, we can refer to recent research in biochemistry [7], neuroscience [75], epidemiology [37] and immunology [104], among others. In this chapter, a number of interesting applications are discussed in detail.

4.1 The Morris-Lecar model in detail

MATCONT was developed for the study of dynamical systems. The reduced Morris-Lecar model, introduced in Section 2.2.3, has been used extensively in many studies. However, no complete bifurcation analysis of the system had ever been published. A limited bifurcation diagram was published in [127], but we found it incomplete and partly incorrect. In this section, a complete bifurcation diagram in the most relevant two-parameter range is constructed using MATCONT, which explains all possible behaviour of the model.

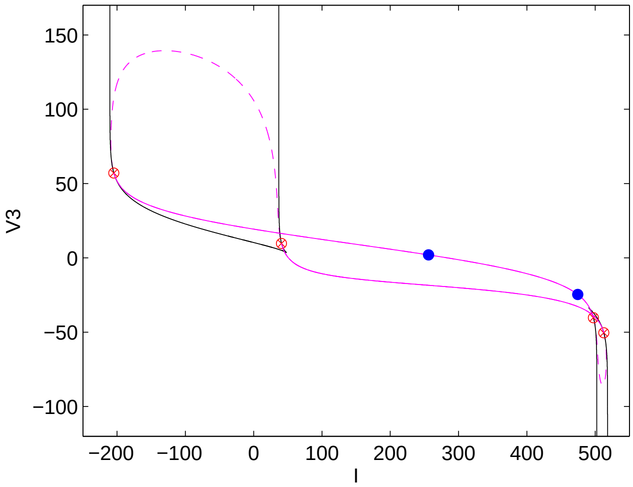


Figure 4.1: Global equilibrium bifurcation diagram of the ML-system. The solid lines are the LP curves, which clearly show CP points. The other line is the Hopf curve. The dotted parts are curves of Hopf points, the dashed parts consist of NS points. The crossed circles indicate the BT points, where both curves touch. The solid discs are GH points.

4.1.1 Bifurcation diagrams

In Figure 4.1 a general view of the equilibrium bifurcation diagram of the Morris-Lecar system for the parameter values listed in Section 2.2.3 is depicted. We also added a qualitative diagram in Figure 4.2. The biologically relevant part is the upper left part, which is enlarged in Figures 4.3 (exact figure) and 4.4 and 4.5 (qualitative diagrams that also include homoclinics and bifurcations of limit cycles).

In Figures 4.1 and 4.2, we see that the biologically relevant range of the system has a mirror-image attached to it: in a very different parameter range, a similar bifurcation diagram shows up again, but ’upside down’. We have noticed that, by varying certain parameters in the system, like bringing the value for C from

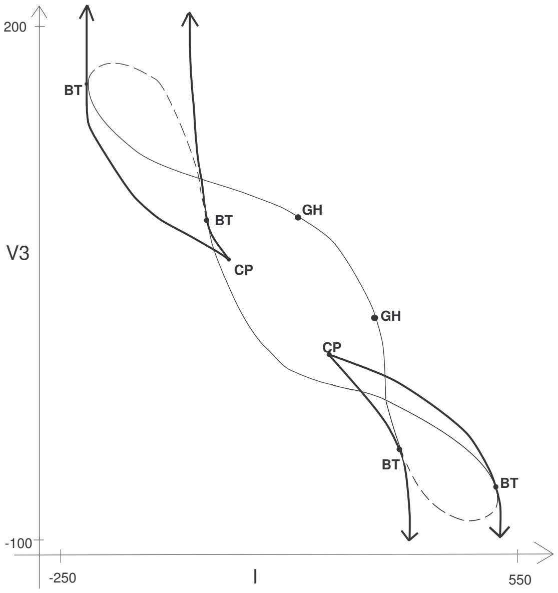


Figure 4.2: Qualitative view of the bifurcation diagram in Figure 4.1. The thickest lines are the LP curves. The dashed line consists of NS points. The thin solid lines are the Hopf curves.

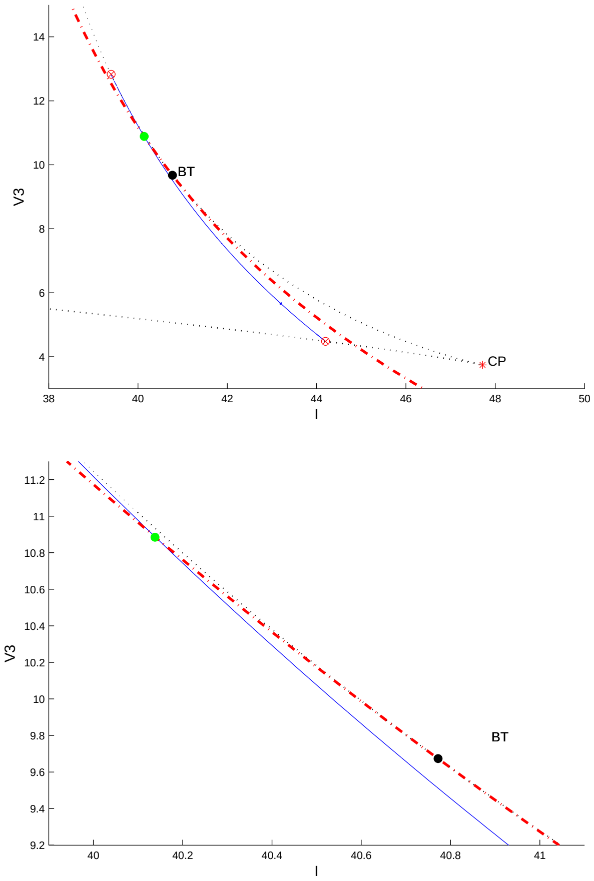


Figure 4.3: Bifurcation diagram of the Morris-Lecar system. The thin solid line is the curve of HHS orbits. The dotted line is the LP curve. The thick dash-dotted line is the curve of Hopf and NS points. There are 4 points marked. The begin- and endpoint of the HHS curve are NCH points. These are marked with a circle with a cross, and are the uppermost and lowermost encircled points in the left picture. The LP curve and Hopf curve touch in the BT point (darkest disc, third circle from the top). The NS curve crosses the curve of HHS orbits in the HNS point (paler disc, second encircled point from the top). The bottom picture is a close-up of part of the top picture.

5 to 20, this ’tail’ or mirror-image in the bottom right part can disappear. The biologically relevant part of the parameter space, however, always remains and preserves its configuration.

4.1.2 Behaviour explained

Equilibria

In Figure 4.4, we subdivide the equilibrium bifurcation diagram into different areas, depending on the number and stability of the equilibria.

In the white areas, there is only 1 equilibrium, and it is stable. In the pale gray area, the unique equilibrium is unstable.

In the other areas there are 3 equilibria, except on the borders: on the LP curves (called L and M) there are exactly 2 equilibria, and in the CP there is only 1 equilibrium.

The dark gray area contains 3 equilibria at each point, of which 1 is stable and 2 are unstable. The 3 equilibria in the horizontally hatched area are all unstable. And finally, the diagonally hatched area has 2 stable and 1 unstable equilibrium.

Limit cycles

Figure 4.5 shows a qualitative diagram of all important bifurcations that occur in the most relevant parameter range for the Morris-Lecar model. Part of it corresponds to Figure 4.3, but we added the bifurcations of limit cycles (notably LPCs) and homoclinics as well as the codimension 2 points related to them.

In Figure 4.6 we present the phase diagrams for the areas as numbered in Figure 4.5. Phase diagrams for the relevant homoclinic bifurcations are shown in Figure 4.7.

We note that both Figures 4.5 and 4.6 are qualitative diagrams. The exact course of the different curves was at times disregarded in favour of clarity in marking the areas with different types of behaviour, and to improve overall clarity of the figures.

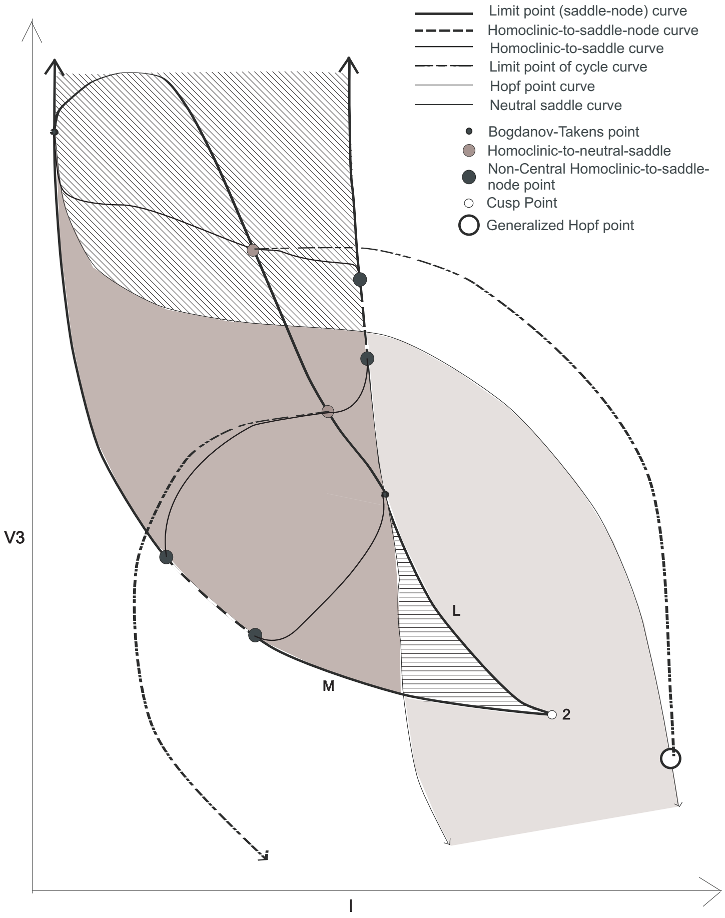

Limit point (saddle-node) curve
Homoclinic-to-saddle-node curve
Homoclinic-to-saddle curve
Limit point of cycle curve
Hopf point curve
Neutral saddle curve
Bogdanov-Takens point
Homoclinic-to-neutral-saddle
Non-Central Homoclinic-to-saddle-node point
Cusp Point
Generalized Hopf point
V3
L
M
2
I

Figure 4.4: Qualitative bifurcation diagram in the biologically most relevant parameter range. Areas are coloured depending on the number and stability of the equilibria: white = 1 stable eq.; light gray = 1 unstable eq.; horizontal lines = 3 unstable eq.; diagonal lines = 2 stable and 1 unstable eq.; dark gray = 2 unstable and 1 stable eq.

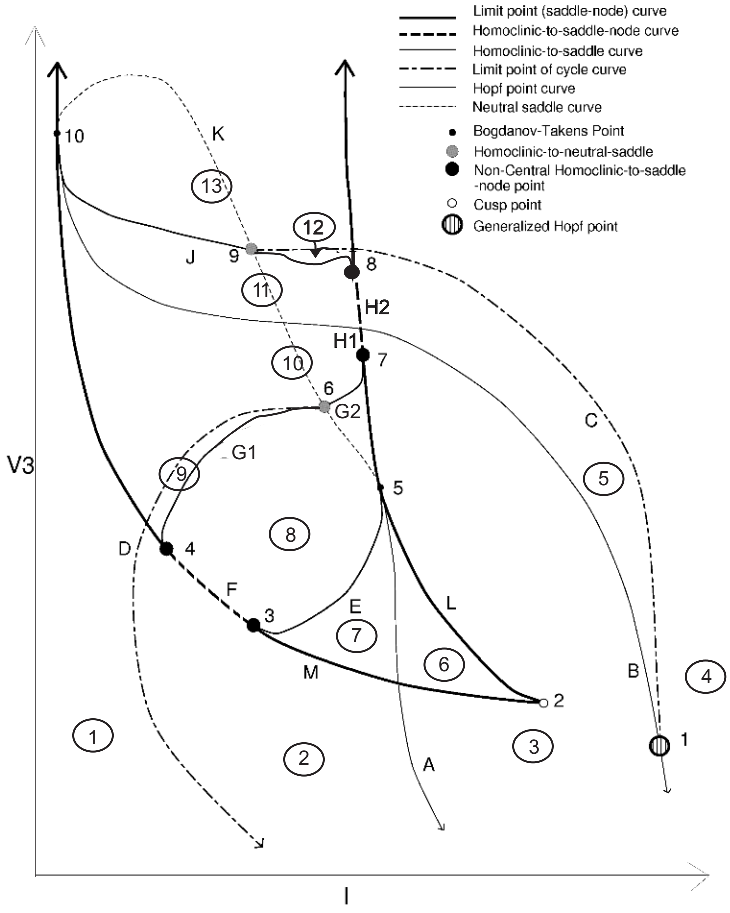

Limit point (saddle-node) curve
Homoclinic-to-saddle-node curve
Homoclinic-to-saddle curve
Limit point of cycle curve
Hopf point curve
Neutral saddle curve
Bogdanov-Takens Point
Homoclinic-to-neutral-saddle
Non-Central Homoclinic-to-saddle -node point
Cusp point
Generalized Hopf point
V3
I
10
K
13
12
J
9
8
11
H2
H1
7
10
6
G2
C
G1
9
5
5
8
D
4
F
E
3
L
7
6
M
B
4
2
1
2
3
1
A

Figure 4.5: Qualitative bifurcation diagram in the biologically most relevant parameter range. V_3-values for the numbered points: $1 = 2.02153$, $2 = 3.74191$, $3 \approx 4.47$, $4 = 4.48019$, $5 = 9.67349$, $6 = 10.88588$, $7 = 12.42132$, $8 \approx 9 \approx 17.25$, $10 = 57.10678$. The encircled numbers refer to phase diagrams for the different plot areas, shown in Figure 4.6.

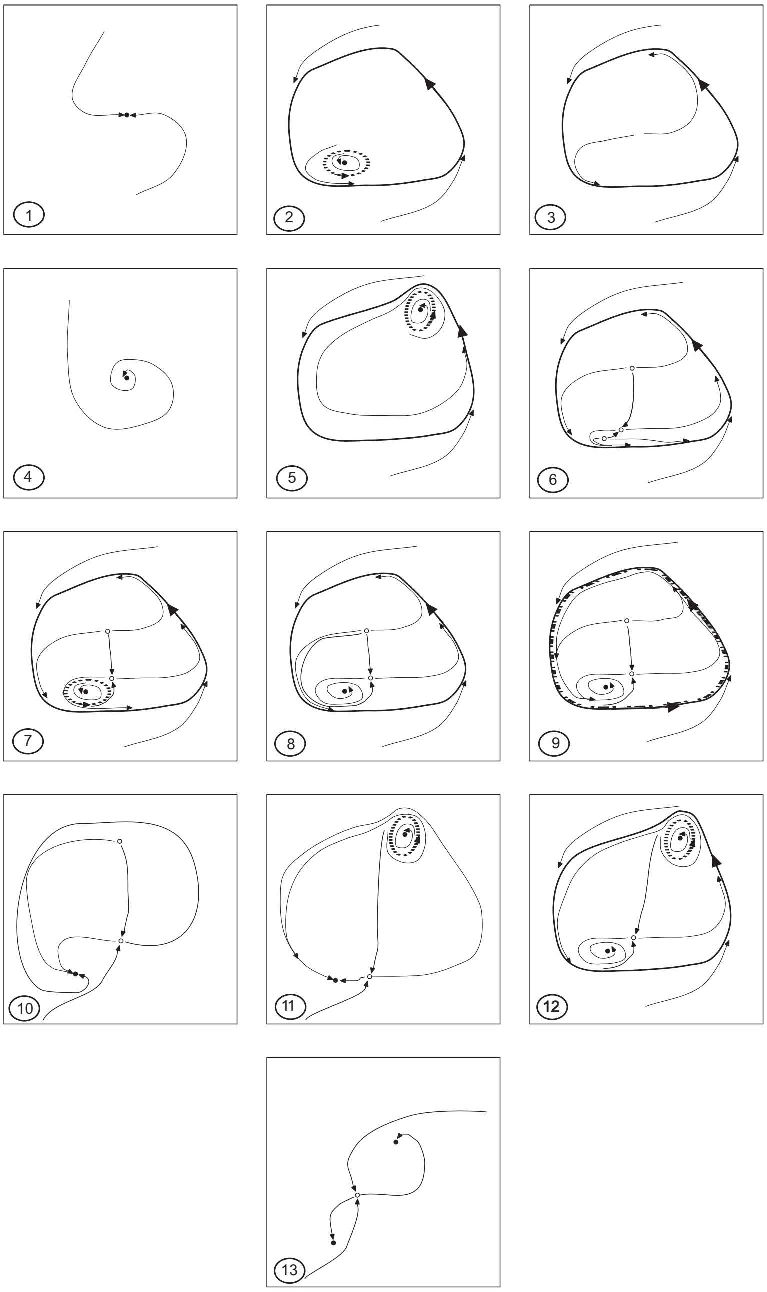


Figure 4.6: Phase diagrams for the areas as numbered in Figure 4.5.

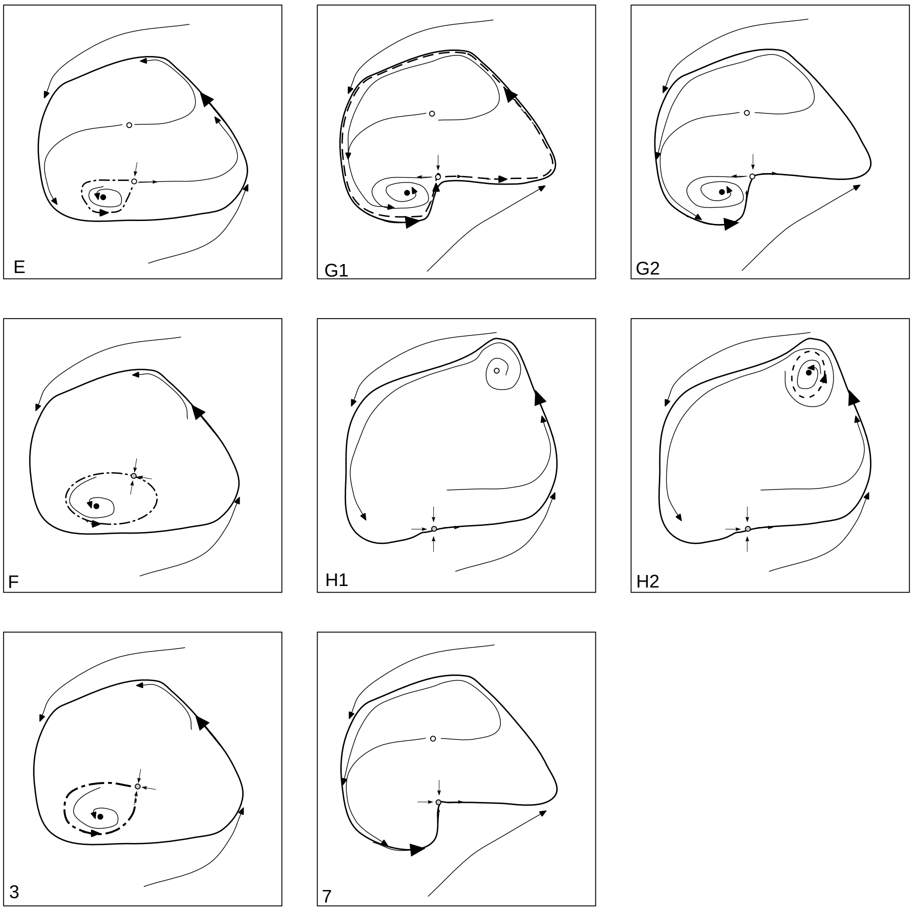


Figure 4.7: Phase diagrams for some homoclinic bifurcations. The letters and numbers in the lower left corners refer to the indications in Figure 4.5. The top 3 show homoclinic-to-hyperbolic-saddle bifurcations; the middle 3 show central homoclinic-to-saddle-node bifurcations; the bottom 2 show non-central homoclinic-to-saddle-node bifurcations.

Classification of types of behaviour

Here we apply the classification, as presented in Section 2.3.3, to the reduced ML model. This classification is summarized in Table 4.1.

In the supercritical parameter range, only two kinds of behaviour of the stable limit cycles are found. Below the GH point 1, so for values $V_3 < V_{3GH} = 2.02153$, the rightmost Hopf point (curve B) consists of supercritical Hopf points: we find behaviour of Group I. For $V_3 > V_{3GH}$, the Hopf points on curve B are subcritical and the stable limit cycles coming from the left, lose stability in a LPC, so we get behaviour of Group II.

In the subcritical range of parameter I, there is no change in behaviour of the stable limit cycles below point 3. In this whole range there is clearly bistability of equilibria and limit cycles, and the firing frequency has a limited range. So we clearly get a Type II bifurcation and Group II behaviour.

From point 3 on, the curves of homoclinics come into play, but with limited effect on the classification. Point 3 forms the transition between HHS and HSN orbits, and is called a Non-Central Homoclinic-to-Saddle-Node (NCH). The name distinguishes these points from the Central Homoclinic-to-Saddle-Node (CHSN) orbits, that are normally found along HSN curves. It is located close to the value of $V_{3HSN1} = 4.47$. Point 4, which is the same type of point, is really close, at the value $V_{3HSN2} = 4.48019$. Between these two points, the unstable limit cycles that arise from the Hopf point on curve A never become stable, but instead end up in a HHS orbit on curve E. The stable limit cycles coming from higher I-values, lose stability at LPC curve D, and then converge to a CHSN orbit on HSN curve F. So the stable limit cycles still cause bistability with the stable equilibrium. After they lose their stability, their period grows to infinity. Since this happens in the unstable regime of limit cycles, this does not influence the actual spiking of the neuron. The firing frequency range remains limited, and we still have Group II behaviour.

Above point 4, the stable limit cycles coming from the right, converge to a HHS bifurcation on curve G1, after losing stability at LPC curve D. This however still does not change the general behaviour.

Point 5 is a BT point, at $V_{3BT} = 9.67349$, and marks the transition between bifurcation Types I and II. For $V_3 > V_{3BT}$, the Hopf points on curve A no longer exist, and the only remaining limit cycles are the ones arising from the Hopf points on curve B. These are still born unstable, gaining stability at the LPC curve C and losing stability at LPC curve D. Thus, when we gradually increase input I, starting low, spiking will start at the limit point, and we clearly get a Type I bifurcation, while there still is bistability, and periods of the stable cycles are still limited in range, which means Group II behaviour.
We note that the classification by equilibrium bifurcation type at the onset of firing (Hopf or LP), as introduced in [110], has no parallel classification in behavioural properties. The bifurcation point, the BT point, does not mark an important transition in firing ranges or onsets.

Also, the Morris-Lecar model shows here, between points 5 and 6, that it is possible to have Group II behaviour at a fold equilibrium bifurcation. This possibility is mentioned in [77] but we are not aware of examples in the literature that exhibit the behaviour.

Point 6 is a HNS point, at $V_{3HNS1} = 10.88588$ and above that point, the LPC curve D doesn't exist anymore. Then the stable limit cycles converge to the HHS orbits on curve G2. So we still get bistability, but firing frequencies can be arbitrarily low, i.e. the periods of the stable cycles can grow to infinity. We have a Type I bifurcation, and subcritical behaviour which is to be classified as Group III.

The next important change happens at NCH point 7, at $V_{3HSN3} = 12.42132$. Here again a transition is made from HHS to HSN orbits. From this point on, there is no more bistability: the stable limit cycles converge to the CHSN on curve H1 or H2. Thus, we get a Type I bifurcation, and Group IV behaviour.

Point 8 is another NCH point at V_{3HSN4}, but the V_3-value of this point, and of point 9, a HNS point, are extremely close together. They both lay somewhere around a V_3-value of 17.30. In the minuscule interval between points 8 and 9, there is again bistability: the stable limit cycles converge to the HHS orbit on

Table 4.1: Summary of classification of the complete biologically relevant parameter range of the reduced Morris-Lecar model.

point range	V_3-range	subcritical behaviour	equilibrium bifurcation	supercritical behaviour
< 1	< 2.02153	Group II	Type II	Group I
$1 - 5$	$]2.02153, 9.67349[$	Group II	Type II	Group II
$5 - 6$	$]9.67349, 10.88588[$	Group II	Type I	Group II
$6 - 7$	$]10.88588, 12.42132[$	Group III	Type I	Group II
$7 - 8$	$]12.42132, 17.30[$	Group IV	Type I	Group II
$8 - 9$	$[17.30]$	Group III	Type I	Group II
> 9	> 17.30	No spiking behaviour		

curve J. Thus, we again have behaviour of Group III.

Point 10 is another BT-point, at $V_3 = 57.10678$. Between points 9 ($V_3 = V_{3HNS2}$) and 10 there are only unstable limit cycles, and above point 10 there aren’t any limit cycles left, so for these parameter ranges the neural model doesn’t show any spiking behaviour.

4.1.3 A broader perspective

The bifurcation diagram discussed here, is not a unique or rare case. An analogous diagram is discussed in [5] (Figure 3.5.6 in Section 3.5.3). It appears also in [76] (see Figure 8) and in numerous other studies.

In fact our bifurcation diagram in Figure 4.5, as well as the diagrams in [5] and [76], are rearrangements of the canonical bifurcation diagrams of Degenerate BT bifurcations with triple and double equilibria [40], [88].

One codimension-3 DBT point with a triple critical equilibrium is found at the intersection of a BT curve and a CP curve (the corresponding parameter values are $V_3 = 117.6322$, $V_1 = 83.4878$ and $I = 228.491$). Location and normal form analysis of codim 3 bifurcations in the present model is an interesting open problem for future research.

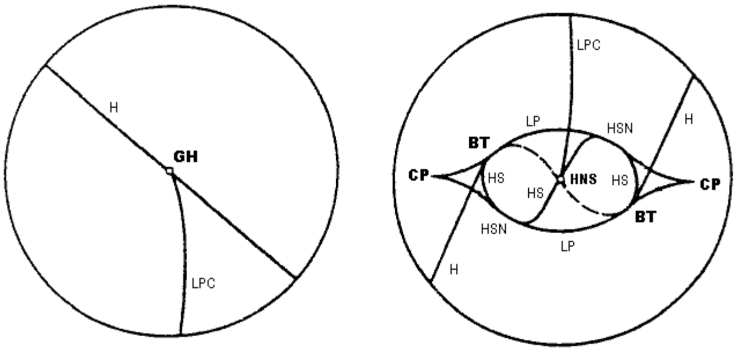


Figure 4.8: Complete bifurcation diagram by Bazykin from [5]. The two circles should be seen as (connected) front and back of a sphere. In boldface, the special points are indicated and in thin lettering, the curves are named.

4.2 Excitatory phase-locking and synchrony

In Section 3.2, we have described how MatCont supports the computation of the PRC and its derivative, the dPRC. In this section, we go deeper into some very interesting applications of these curves, having to do with phase-locking and synchrony of neurons.

We recall that the phase response curve was originally introduced as:

$$PRC(t, I) = \frac{T_{old} - T_{new}}{T_{old}} \;, \qquad (4.1)$$

where T_{old} is the period of the unperturbed cycle, and T_{new} is the time span between the previous spike and the next spike when the input pulse I is given at time t. In this section, we will usually write the phase response curve as $\phi(t) = PRC(t, I)$.

We start with the possibility of getting a neuron to phase-lock with a repeated input, then we study the case of a 2-neuron network with excitatory connections, and finally extend this theory to an N-neuron network.

4.2.1 1:1 Phase-locking to repeated inputs

Consider an oscillator with period T, which gets repeated external inputs with a fixed period which is close to T, say T_Δ. The ability of the oscillator to phase-lock with this input is an important concept, as discussed in [13] and [78].

Say the oscillator or neural model fires a spike at time $t = 0$, and gets an input with strength α at time $t = t_1$. Then the next spike of the neuron will occur at time $t = T - \alpha T\ \phi(t_1)$. If we call t_2 the distance between the following spike and input, then this can be computed as the difference between the time of the input and the time of the spike:

$$t_2 = (t_1 + T_\Delta) - T(1 - \alpha\phi(t_1))\ . \tag{4.2}$$

So

$$t_2 = (t_1 + T_\Delta) + \alpha T\ \phi(t_1)\ \texttt{mod}\, T\ . \tag{4.3}$$

Thus, we can consider the time difference map between spike and input:

$$F_{diff} :\ \ t_1 \to t_1 + T_\Delta + \alpha T\phi(t_1) - T\ . \tag{4.4}$$

Any fixed point of this map has the property that

$$T_\Delta + \alpha T\ \phi(t_1) = 0\ \texttt{mod}\, T\ . \tag{4.5}$$

In these points, 1:1 phase locking occurs. Since the linearized PRC of a neuron can only be used reliably for α small enough, the period of the inputs should not be too different from the oscillator’s original period.

You can interpret (4.5) as follows: when the PRC ϕ for a certain delay is positive (negative), then the next spikes will be sped up (delayed), therefore there can only be synchronization if $T_\Delta < T$ ($T_\Delta > T$).

The so obtained fixed point of the map (4.4) is stable if and only if its unique multiplier $1 + \alpha T\phi'(t_1)$ has modulus less than 1, i.e.

$$\frac{-2}{\alpha T} < \phi'(t_1) < 0\ . \tag{4.6}$$

So, if an oscillator gets a series of inputs with a period close to its own original period (T_Δ satisfies (4.5)), and the delay t_1 meets constraint (4.6), then one can

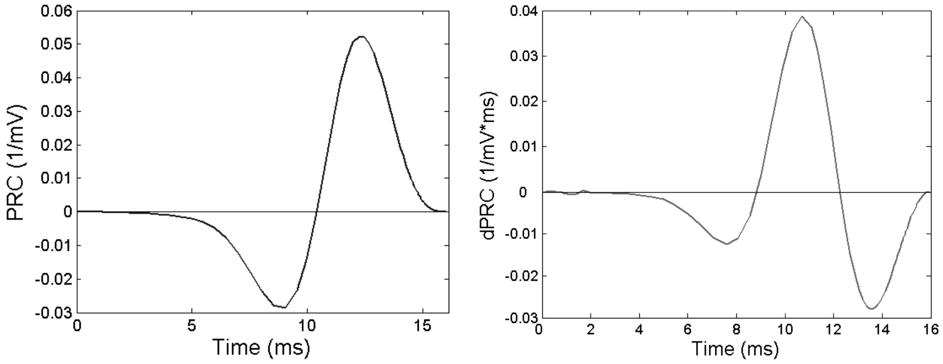


Figure 4.9: Left: PRC (ϕ) for the Hodgkin-Huxley system. Right: dPRC (ϕ') for the same system.

get stable phase-locking of the oscillator to the inputs.

These results were tested on the Hodgkin-Huxley equations, as described in Section 2.13, at the parameter values listed there, and $I = 7.996$, with inputs of strength 1, for which the PRC and dPRC are shown in Figure 4.9. At these parameter values, the period of the system is around 16.12. For an input with a period larger than T, both ϕ and ϕ' will have to be negative at the input time to allow phase-locking, so the inputs will have to come earlier than about 8.5 ms after the last spike. In all cases with negative ϕ and negative ϕ', $\phi' > -2/T = -0.125$, so constraint (4.6) is always satisfied. And finally, the exact timing of input and spike at the phase-locked state will depend on the period of the spike. Since $T_\Delta > T$, we know that $T_\Delta - T = T\ \phi(t_1)$.

E.g. when we give inputs with period $T_\Delta = T(1 + 0.01) \approx 16.28$, the neuron will always phase-lock with these inputs if they reach the neuron at time t_1, such that $\phi(t_1) = -0.01$ and $\phi'(t_1) < 0$, this is the case for $t_1 = 6.747$ ms. The results of a numerical simulation are shown in Figure 4.10, with circles indicating the times of arriving input. An analogous analysis can be done for inputs arriving with a shorter period than that of the uncoupled model ($T_\Delta < T$).

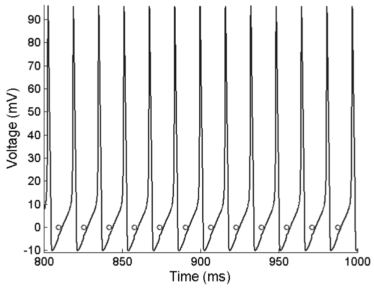


Figure 4.10: The spikes show the spiking regime of the Hodgkin-Huxley neuron. The circles indicate when the repetitive input comes in. It is clear that this is a phase-locked state, for $t_1 \approx 7$ ms, with period 16.28 ms.

4.2.2 Two single-spiking neurons

Synchrony

Consider two identical neurons with excitatory connections to each other, and suppose that the connection has a delay of duration t_1 and a connection strength of size α. Suppose the neurons, without input, have period T, and the linearized PRC of the neurons is given by the function ϕ. Finally assume that neuron 1 fires at time $t = 0$, and neuron 2 at time $t = \Delta$, $\Delta < t_1$.

Then neuron 1 gets an input signal at time $\Delta + t_1$ and neuron 2 gets a signal at absolute time t_1, which means, relative to its spike, at time $t_1 - \Delta$ (due to the phase difference).

For the next spikes we conclude:
- neuron 1 will fire its next spike at $t = T - \alpha T\ \phi(t_1 + \Delta)$.
- neuron 2 will fire its next spike at $t = (\Delta + T) - \alpha T\ \phi(t_1 - \Delta)$.

The time difference map (for the time difference between spikes of the two neurons)

$$\Delta \to \Delta + \alpha T \phi(t_1 + \Delta) - \alpha T \phi(t_1 - \Delta) \tag{4.7}$$

has the trivial fixed point $\Delta = 0$, which corresponds to synchrony.

The right-hand side of (4.7) can be simplified:

$$\begin{aligned} &= \quad \Delta + \alpha T(\phi(t_1 + \Delta) - \phi(t_1 - \Delta)) \\ &\approx \quad \Delta + 2\alpha\Delta T\phi'(t_1) \end{aligned} \tag{4.8}$$

The fixed point $\Delta = 0$ of map (4.7), and thus the synchrony, is stable if and only if the unique multiplier of the map, $1 + 2\alpha\phi'(t_1)$, has modulus less than 1, i.e.

$$\frac{-1}{\alpha T} < \phi'(t_1) < 0 \; . \tag{4.9}$$

Once the neurons are synchronized, the new period is $T - \alpha T \; \phi(t_1)$.

This shows that for two identical neurons with excitatory connections to synchronize, the connection delay plays a crucial role. Moreover, the delays that allow synchronization, can be determined solely from the period of the uncoupled neuron, its PRC and its dPRC.

Consider for example two identical Hodgkin-Huxley neurons, connected by excitatory connections with strength 1. From the plot of ϕ' in Figure 4.9, we notice that $\phi'(t_1)$ is always greater than $-1/T \approx -0.0624$. So constraint (4.9) is always met when $\phi'(t_1) < 0$. So whenever we initiate the neurons with a connection delay t_1 such that $\alpha \leq 1$, $\phi'(t_1) < 0$, the neurons will move to synchronization with new period $T - \alpha T \; \phi(t_1)$.

Results from an example simulation are shown in Figure 4.11. The first plot shows the start, and the second one the result, when starting with 2 HH-neurons, firing 1 ms apart, and with a connection delay of 6 ms. They clearly converge, with a new period of 16.1 ms, which matches our formula.

Analogously, experiments support the theory that there is synchronization of the two neurons with a shorter emerging period when $\phi'(t_1) < 0$ and $\phi(t_1) > 0$, and there is no synchronization if $\phi'(t_1) > 0$.

1:1 phase-locking

If a neuron has a phase response curve ϕ such that $\phi(t_1 + \Delta) = \phi(t_1 - \Delta)$, for a certain connection delay t_1 and a phase difference Δ, then this point is also a

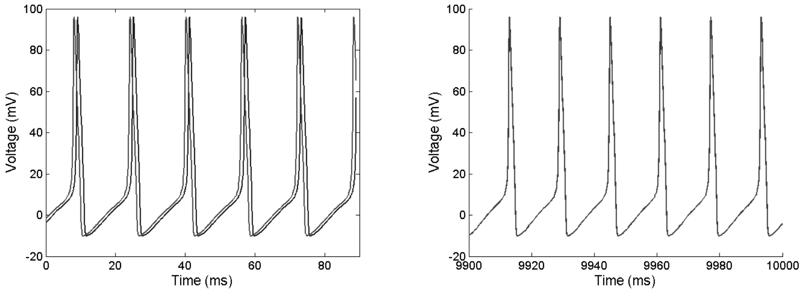


Figure 4.11: Left: 2 HH-neurons spiking with 1 ms difference in timing. Right: The 2 neurons have converged.

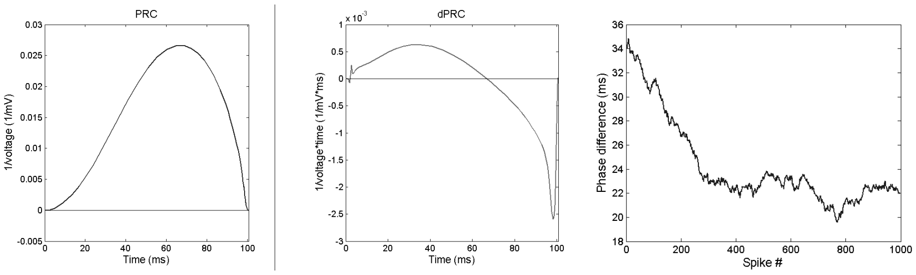


Figure 4.12: Left: PRC for the Connor model. Middle: dPRC for the same model. Right: phase difference between two spiking neurons with excitatory connection, evolving to a stable state.

fixed point of the map (4.7).

This fixed point is stable when

$$
\begin{aligned}
|1 + \alpha T \phi'(t_1 + \Delta) + \alpha T \phi'(t_1 - \Delta)| &< 1 \\
\frac{-2}{\alpha T} < \phi'(t_1 + \Delta) + \phi'(t_1 - \Delta) &< 0\, . \qquad (4.10)
\end{aligned}
$$

So the fall of ϕ must be steeper than its rise.

As an example, this is the case in the Connor model. The equations for this model are given in Section 2.2.2. Figure 4.12 shows the PRC (left) and dPRC (middle) of the Connor model at parameter value $I = 8.49$. When the connection delay $t_1 = 64$, then $\Delta \approx 22$ is a fixed point, such that $\phi(t_1 + \Delta) = \phi(t_1 - \Delta) = 0.019$. Also $\phi'(t_1 + \Delta) = -8e - 4$ and $\phi'(t_1 - \Delta) = 6e - 4$, so constraint (4.10) is met.

A simulation result is shown in the rightmost picture in Figure 4.12; it shows how the phase difference between two spiking neurons moves from its initial value, chosen at 35 ms, to 22 ms, the stable fixed point.

4.2.3 Synchronization of an N-neuron single-spiking network

It is possible to generalize the theory for a 2-neuron network to an N-neuron network. Consider a network of N identical oscillators with period T and PRC ϕ. Suppose this is an all-to-all connected excitatory network, with all connections having a delay t_1 and strength α.

Let $X = (x_1\ x_2\ ...\ x_N)^T$ be a vector of time variables close to each other such that oscillator 1 spikes at time x_1, oscillator 2 at time x_2, etc. Then the next spikes will be fired at times

$$A(X) = \begin{pmatrix} x_1 + T - \alpha T\phi(t_1 + x_2 - x_1) - ... - \alpha T\phi(t_1 + x_N - x_1) \\ x_2 + T - \alpha T\phi(t_1 + x_1 - x_2) - ... - \alpha T\phi(t_1 + x_N - x_2) \\ ... \\ x_N + T - \alpha T\phi(t_1 + x_1 - x_N) - ... - \alpha T\phi(t_1 + x_{N-1} - x_N) \end{pmatrix}. \tag{4.11}$$

If there is synchronization, i.e. $x_1 = x_2 = ... = x_N$, then all components of $A(X)$ are equal to $x_1 + T - \alpha(N-1)T\phi(t_1)$. So the period of the synchronized state must be $T - \alpha(N-1)T\phi(t_1)$.

We want to know whether this synchronized state is stable and attracting. Consider the map

$$F : X \mapsto A(X) - \begin{pmatrix} T(1 - \alpha(N-1)\phi(t_1)) \\ T(1 - \alpha(N-1)\phi(t_1)) \\ ... \\ T(1 - \alpha(N-1)\phi(t_1)) \end{pmatrix}. \tag{4.12}$$

F has a fixed point $(0 \,...\, 0)^T$ and its Jacobian F_X evaluated there is

$$\begin{pmatrix} 1+\alpha T\phi'(t_1)(N-1) & -\alpha T\phi'(t_1) & ... & -\alpha T\phi'(t_1) \\ -\alpha T\phi'(t_1) & 1+\alpha T\phi'(t_1)(N-1) & ... & -\alpha T\phi'(t_1) \\ ... & ... & ... & ... \\ -\alpha T\phi'(t_1) & ... & ... & 1+\alpha T\phi'(t_1)(N-1) \end{pmatrix} . \tag{4.13}$$

To determine the asymptotic stability of the synchronized state, we consider the difference vector $Y = (y_2 \,...\, y_N)^T = (x_2 - x_1 \,...\, x_N - x_1)^T$. The difference vector of $F(X)$ is $G(Y) = (F_2(X) - F_1(X) \;\, ... \;\, F_N(X) - F_1(X))^T$ where F_i is the i - th component of F.

Now the stability of the synchronized state is determined by the (multipliers of the Jacobian of the) map G. Its Jacobian G_Y can easily be obtained from F_X and is given by

$$\begin{pmatrix} 1+\alpha T\phi'(t_1)(N-1) & ... & -\alpha T\phi'(t_1) \\ ... & ... & ... \\ -\alpha T\phi'(t_1) & ... & 1+\alpha T\phi'(t_1)(N-1) \end{pmatrix} + \begin{pmatrix} \alpha T\phi'(t_1) & ... & \alpha T\phi'(t_1) \\ ... & ... & ... \\ \alpha T\phi'(t_1) & ... & \alpha T\phi'(t_1) \end{pmatrix} . \tag{4.14}$$

This is a diagonal matrix with $(N-1)$ equal diagonal elements and thus all eigenvalues equal to $1+\alpha N\, T\phi'(t_1)$. The fixed point $(0 \,...\, 0)^T$ of the map F is asymptotically stable if all multipliers are strictly inside the (complex) unit circle. Thus the synchronization is asymptotically stable if

$$\begin{aligned} -1 < 1 + \alpha N\, T\phi'(\delta) < 1 \\ -2 < \alpha N\, T\phi'(\delta) < 0 \\ \frac{-2}{\alpha N\, T} < \phi'(\delta) < 0 \,. \end{aligned} \tag{4.15}$$

This is the generalization of constraint (4.9) to the synchronization of an N-neuron network with all-to-all excitatory connections.

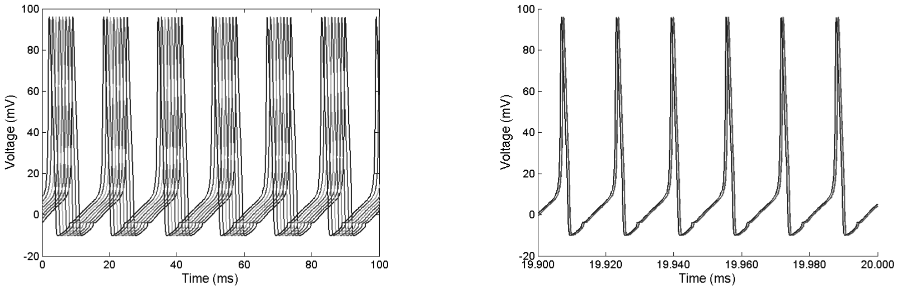


Figure 4.13: Left: 8 HH-neurons spiking with 1 ms difference in timing. Right: The 8 neurons have converged.

This is again confirmed by tests on a network of Hodgkin-Huxley neurons. Figure 4.13 shows on the left the initial state: 8 HH neurons, firing spikes 1 ms apart. They are connected through excitatory connections, giving each other pulses of size 1 (mV). The connections have a delay of 6 ms. After 20000 ms, the result is shown on the right of the figure: all 8 neurons are strongly synchronized.

4.3 Further applications of PRCs in MATCONT

In this section we combine MATCONT’s functionalities of continuation and computing the PRC, to obtain families of PRCs in cases where the shape of the PRC is known to have important consequences on networking behaviour. The change of the shapes of the PRCs under parameter variation therefore is an interesting object of study.

4.3.1 Morris-Lecar: limit cycles close to homoclinics

Brown et al. [13] have studied the response dynamics of weakly coupled or uncoupled neural populations using the PRCs of periodic orbits near certain bifurcation points. In particular they obtain the PRCs of the periodic orbits near homoclinic-to-saddle-node orbits (HSN; Brown et al. call this the SNIPER bifurcation) and homoclinic-to-hyperbolic-saddle orbits (HHS; Brown et al. call this the homoclinic bifurcation). They obtain standard forms for these PRCs, based on a normal form analysis of the homoclinic orbits.

In our numerical calculations that involve the Hindmarsh-Rose model in the first (HSN) case and the Morris-Lecar model in the second (HHS) case, their normal-form-based predictions are largely confirmed.

It turns out that the PRCs in the two cases look very different. On the other hand, it is well known that the transition from HSN to HHS orbits is generic when a curve of HSN or HHS orbits is computed. The generic transition point is a non-central homoclinic-to-saddle-node orbit (NCH). Moreover, this transition is not uncommon in neural models and indeed we found it in the ubiquitous Morris-Lecar model, cf. Section 4.1.

So one can conclude that the PRCs near HSN orbits can smoothly be transformed into the PRCs near HHS orbits. Analytically this calls for a normal form analysis of the NCH orbits. We have not done this, but used the situation as a testing ground for our computation of PRCs.

We computed a branch of spiking orbits with fixed high period (this forces the branch to follow the homoclinic orbits closely) from the HHS regime into the HSN regime, computed the PRCs in a large number of points and plotted the resulting PRCs in a single picture to get a geometric insight in the way the PRCs are transformed from one into the other.

In Figure 4.14A, part of the phase plane is shown for the Morris-Lecar model. Since the curves are very close together, we added a qualitative diagram (Figure 4.14B). The pictures show the saddle-node curve (thin), and the curve of HHS orbits (dashed), that intersect and collapse to form a curve of HSN orbits (thick). The point of intersection is encircled; this is a NCH point. Close to the curves of HHS and HSN orbits, is the curve (dotted) of limit cycles with fixed period (237.665) along which we have computed PRCs. The continuation was done using 80 mesh intervals and 4 collocation points per interval, which results in high precision computations.

Figure 4.14C shows the resulting 100 computed PRCs. We started the limit cycle continuation from a cycle with $V_3 = 10$ and $I = 40.536$. The corresponding

PRC is the one that, in Figure 4.14C, has the leftmost peak; it is also slightly darker in the picture than the following PRCs. The picture shows the smooth transition of consecutive PRCs. The gray PRCs are the ones for limit cycles at values of V_3 lower than the value at the NCH point ($V_3 = 12.4213$). The dark PRCs correspond to limit cycles at higher V_3-values.

The shapes obtained for PRCs near HSN orbits and those near HHS orbits both confirm the results from [13].

Two significant facts come out from the picture. First, the dark PRCs are (at least close to) strictly positive, while the gray PRCs have a more distinct negative region. Secondly, the PRCs closest to the V_3-value of the NCH point are also the PRCs with the lowest peak, i.e. the PRCs in the bottom of the ’valley’ that is formed by the consecutive PRCs for limit cycles further away from that particular value, in either direction. This suggests that the NCH orbit has a distinct influence on the shape of the PRCs of nearby limit cycles. This is not surprising, but to our knowledge has never been investigated.

The computation of the 100 PRCs along the limit cycle curve took a total time of 2.34 s, i.e. 0.0234 seconds per PRC. To compare this with standard methods, we note that in [46] §9.5.3, Ermentrout states that to compute one PRC takes a second or two. This is certainly acceptable if one is interested in a single PRC but hardly acceptable if one is interested in the evolution of the PRC under a change of parameters.

The full continuation run for these 100 limit cycles took 62.41 seconds. So our PRC-computations took only about 3.75% of the total time needed for the experiment. If one would use a PRC-computing method that takes for example 2 seconds per PRC, this would cause the total runtime of the program to increase up to 260 seconds, an increase of more than 300%.

4.3.2 Hodgkin-Huxley: limit cycles close to saddle-node of cycles

As a third test, we have done a continuation of both coexisting stable limit cycles in the Hodgkin-Huxley system, mentioned in Section 3.2.5. In both cases we did the continuation for decreasing values of I, approaching a saddle-node of limit

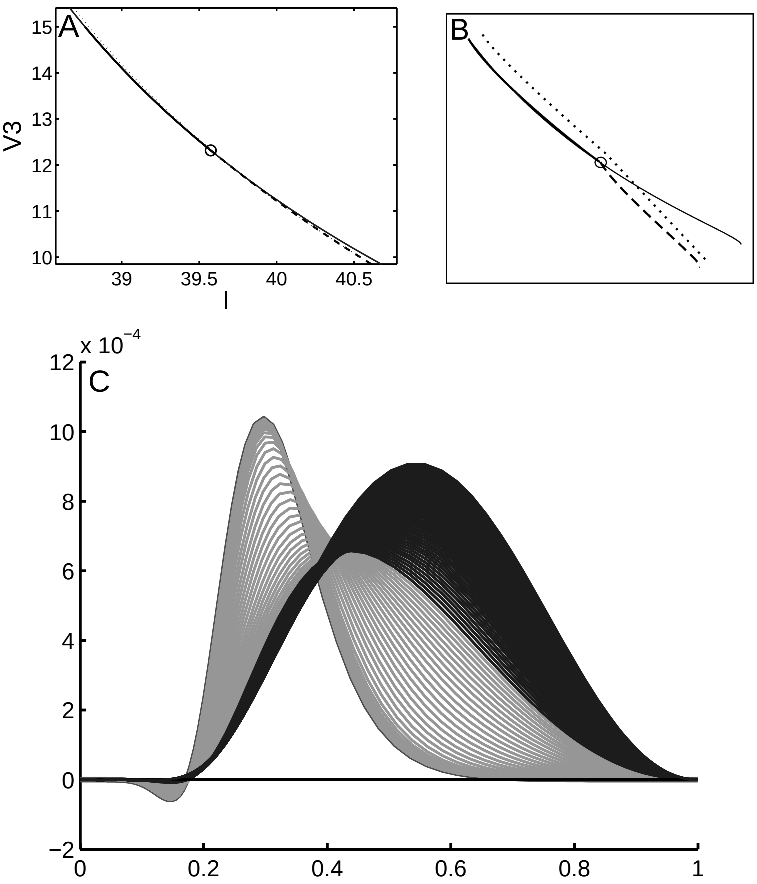


Figure 4.14: A) Phase plot for the Morris-Lecar model. B) Qualitative picture to clarify relative positions of the curves. Thin = limit point curve; thick = HSN-curve; dashed = HHS curve; circle = NCH point; dotted = curve of limit cycles with period $237,6646$. C) PRCs for limit cycles along limit cycle curve. Gray = PRCs for limit cycles close to HHS curves; black = PRCs for limit cycles close to HSN curves.

cycles (LPC).

The big stable limit cycle loses its stability at an LPC that occurs at $I = 6.276$. Starting from $I = 7.919$, we computed 150 continuation steps, with 80 mesh intervals and 4 collocation points. The final computed limit cycle was at $I = 6.377$. The last 100 PRCs are shown in Figure 4.15A, where the palest one, with biggest amplitude, is the PRC corresponding to the limit cycle closest to the LPC.

The small stable limit cycle loses its stability at an LPC that occurs at $I = 7.8588$. Starting from $I = 7.919$, we computed 100 continuation steps, with smaller stepsizes than for the big limit cycle. The final computed limit cycle was at $I = 7.8592$. The corresponding 100 PRCs are shown in Figure 4.15B; again the palest one is closest to the LPC.

During the continuations, the actual PRC-computations took around 4 seconds total for 100 PRCs, i.e. 0.04 s per PRC. The full continuation run took around 100 seconds, so the PRC-computations took about 4% of the total runtime. Again, the use of another method to compute the PRC, that takes 2 seconds per limit cycle, would increase the time needed to about 300 seconds, an increase by 200%.

Both PRCs have both positive and negative regions but otherwise their shapes are very different. It is noteworthy that the shapes of the PRCs near the big LPC look similar to the PRCs predicted near the Bautin bifurcation in [13]. It is well known that branches of LPCs generically are born at Bautin bifurcation points. However, the shapes of the PRCs near the small LPC are very different. We note that both LPCs are far away from Bautin bifurcation points. This is again a subject for further investigation.

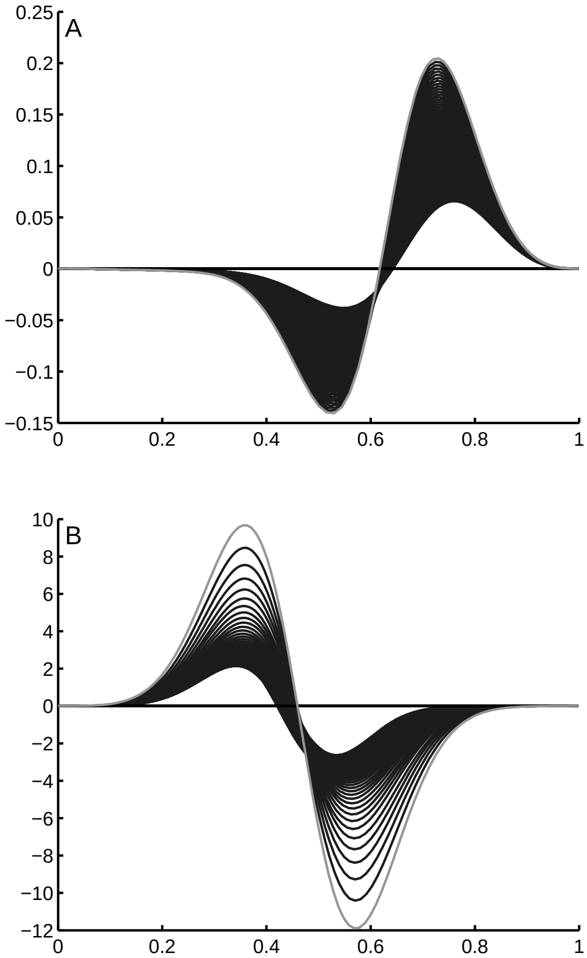


Figure 4.15: A) PRCs of big Hodgkin-Huxley limit cycles approaching a LPC at $I = 6.276$. B) PRCs of small Hodgkin-Huxley limit cycles approaching a LPC at $I = 7.919$.

Chapter 5

Neuron and synapse modelling in the young *Xenopus* Tadpole

Humor can be dissected as a frog can, but the thing dies in the process and the innards are discouraging to any but the pure scientific mind.
(E. B. White (1899 - 1985), Some Remarks on Humor, introduction)

In many neuronal circuits, in both invertebrates and vertebrates, different neurons have rather different properties. However, particularly in complex vertebrate central nervous systems, the normal activity of neurons during behaviour is difficult to study. Using a new whole-cell patch recording method [94] in the spinal cord of immobilized, newly hatched Xenopus tadpole, people from the *Xenopus* tadpole research group of the University of Bristol have been able to make measurements of the type-specific properties of individual neurons, the synapses they make, and their activity during defined motor responses. When we started modelling work on this young *Xenopus* tadpole, this data put us in a nearly unique position for a vertebrate, where we can make models of each individual type of neuron and synapse on the basis of available physiological data. In this chapter detailed information is given on all our individual neuron and synapse models, which will subsequently, in the next chapter, be used in several network models.

Figure 5.1: Chain of full-grown *Xenopus laevis*. (Picture from the Cohen-Cory lab website, http://cohen-corylab.bio.uci.edu/)

5.1 Introduction to the *Xenopus* tadpole

5.1.1 The *Xenopus* tadpole

The *Xenopus laevis* or African Clawed Frog is an animal that exemplifies how nature can be hesitant to modify successful adaptation. Fossils of the *Xenopus* have been found, dating from the Cretaceous (144 to 65 million years ago) [116], and the animal is nowhere endangered, even today [125]. The *Xenopus laevis* was the first vertebrate to be successfully cloned [48], and is a very popular animal to study early vertebrate properties.

The hatchling *Xenopus* tadpole provides a very simple model animal with limited behaviour. It can swim either spontaneously or when touched anywhere on the body [11]. A pineal eye detects dimming light, which speeds up swimming [79]. The tadpole stops swimming when the head bumps into solid objects and it sticks to things with mucus secreted by a cement gland on the head [11]. If the tadpole is held, it can make stronger struggling movements [123]. Swimming

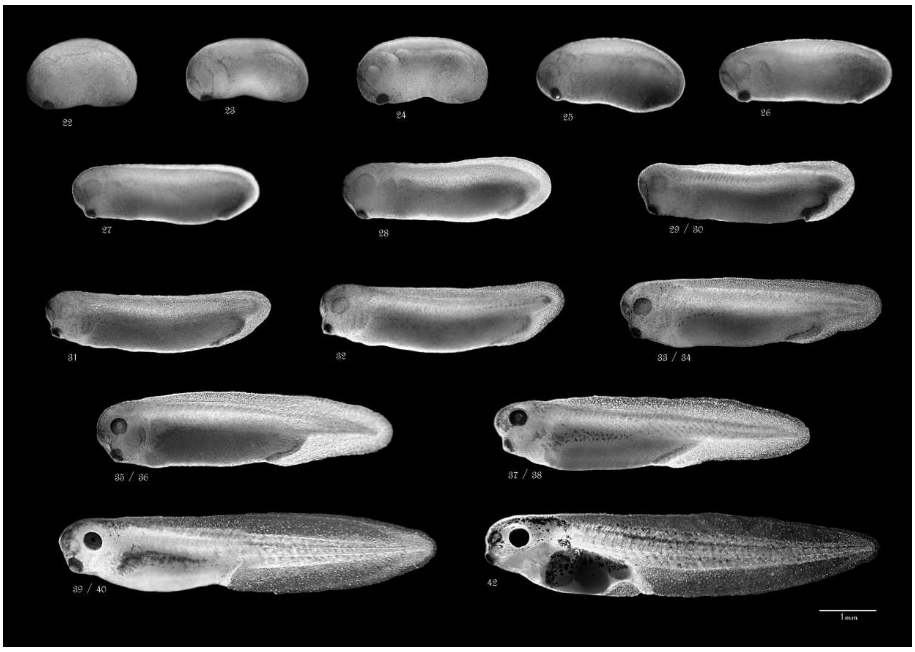


Figure 5.2: Initial developmental stages of the *Xenopus laevis*. (Picture from the *Xenopus* tadpole research group at the University of Bristol)

behaviour in particular has been studied in detail at the neuronal level in physiological experiments ([121],[111],[25],etc.). Of the struggling behaviour, much less is known.

The behaviour of the hatchling *Xenopus* is being studied in the hope of getting a better grasp on how nervous systems, even simple ones, are organized on the cellular level. The studies comprise a combination of techniques: behavioural observation, whole cell recording and network modelling. The present chapter and the next one deal precisely with the modelling aspect of this research.

At the University of Bristol, the *Xenopus* tadpole research group has been

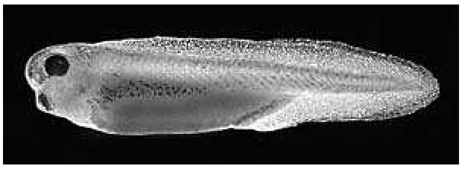


Figure 5.3: Hatchling *Xenopus laevis* at stage 37/38. (Picture from the *Xenopus* tadpole research group at the University of Bristol)

studying this little creature for many years. Professor Alan Roberts leads this group, and for two years, I have been involved in the effort of modelling the nervous system of the animal, based on the measurements and data from the group. The research group has been focussing on one specific stage in the development of the hatchling *Xenopus* tadpole, namely stage 37/38 [102]. A series of these developmental stages are shown in Figure 5.2, and a close-up of a tadpole at stage 37/38 is given in Figure 5.3.

5.1.2 Neuron types in the young *Xenopus* tadpole

The spinal cord of vertebrates has a range of different neuron types with different properties. In some simpler cases like the lamprey [105], [63], and the early developmental stages of the zebrafish [70], [71] and frog *Xenopus* [115] there is actual detailed information on the neuron types and their activity during swimming. In lamprey and *Xenopus* this has allowed modelling of the spinal networks underlying the generation of swimming activity and the testing of hypotheses on the neurons and cellular mechanisms responsible [84], [128]. So far, while some modelling has introduced stochasticity [69], it has not addressed the significance of the different properties of the component neuron types, in part because detailed data were not available.

From detailed measurements, done in the *Xenopus* tadpole research group in the University of Bristol, using a new whole-cell patch recording method [94] in the spinal cord of immobilized, newly hatched Xenopus tadpole, data on the type-specific properties of individual neurons, the synapses they make, and their activity during defined motor responses was gathered.

Previous models of the hatchling *Xenopus* tadpole spinal swimming circuits have assumed that all CPG neurons have the same properties [114], [24], [128], [26]. The data from the recent recordings show that this assumption is wrong: different neuron types show clearly distinct properties.

The newly hatched *Xenopus* tadpole, with only a couple of thousand neurons in the spinal cord, provides a suitable system to study the type-specific differences between neurons. These neurons can be divided into 8 distinct classes on the basis of clear anatomical differences: dorsoventral soma position (i.e. relative position from back to belly of the animal), dendrite distribution and axonal projections; [115], [89]. Nothing is known about the function of one class of possible neural canal receptor neuron [23]. For the remaining neuron classes, using single-cell recordings in the spinal cord of immobilized tadpoles, we have defined the type-specific properties of individual neurons whose anatomical features were revealed by neurobiotin injection. These include: their passive and active responses to current injection, the transmitter pharmacology of synapses they make and receive (where some synaptic conductances have been studied in voltage-clamp recordings), and their activity during responses to sensory stimuli. Such detailed information has shown that 7 of the anatomical classes correspond to distinct functional classes: 1 type of sensory neuron innervating the skin [18], [91]; 2 types of sensory pathway neuron that have a role in the initiation of swimming [91], [93]; 3 types of premotor neuron that are active during swimming and may be components of the swimming central pattern generator [22], [90], [95], [96]; and motoneurons [121], [113]. These detailed studies have recently provided evidence of functional sub-types within two anatomical classes (Li et al., unpublished). We should emphasize that, apart from the primary sensory neurons [109], there is presently no evidence on the ionic channels that different specific types of *Xenopus* spinal neurons possess.

Table 5.1: Different types of neuron in the hatchling *Xenopus* spinal cord and their functions: sensory neurons and sensory pathway neurons.

Neuron type	Abbrev.	Function and position
Rohon-Beard cell	RB	Sensory neuron that innervates the skin, and excites dlc and dla. Located dorsally.
Dorsolateral ascending interneuron	dla	Sensory pathway interneuron, that excites ipsilateral CPG neurons (= central pattern generator neurons on the same side). Located dorsally. They can get inhibited (blocked) by the ipsilateral aINs (= aINs on the same side). It is inhibited during swimming and struggling.
Dorsolateral commissural interneuron	dlc	Sensory pathway interneuron, that excites contralateral CPG neurons (= CPG neurons on the opposite side). Located dorsally, with axons that cross the spinal cord ventrally. They can get inhibited (blocked) by the ipsilateral aINs. It is inhibited during swimming and struggling.

Table 5.2: Different types of neuron in the hatchling *Xenopus* spinal cord and their functions: CPG (= central pattern generator) neurons.

Neuron type	Abbrev.	Function and position
Descending interneuron	dIN	CPG neuron that excites all ipsilateral CPG neurons. Located rather dorsally. Active during swimming, inactive during struggling. Fires only single spikes.
Ascending interneuron	aIN	CPG neuron that inhibits all ipsilateral CPG neurons, dla and dlc. Located in the dorsoventral middle of the tadpole. Fires single spikes during swimming, and bursts of spikes during struggling.
Commissural interneuron	cIN	CPG neuron that inhibits all contralateral CPG neurons. Located in the dorsoventral middle of the tadpole, with axons that cross the spinal cord ventrally. Fires single spikes during swimming, and bursts of spikes during struggling.
Motoneuron	MN	CPG neuron that connects to ipsilateral muscles and causes their activity. Located ventrally. Fires single spikes during swimming, and bursts of spikes during struggling.

5.2 Individual neuron models

5.2.1 General equations

To begin the modelling of individual neurons, we started from previous models built specifically for *Xenopus* spinal neurons [114], [24], [25]. We used a single compartment model since these small neurons have very short dendrites and, if the axon is ignored, are electrically compact [25], [130]. Some parameters were taken directly from experimental data on individual neuron types (e.g. leak conductance and reversal potential). However, since membrane currents have not yet been characterized for the different neurons types, in most cases we used 'emerging' parameters obtained by matching outputs of the model to physiological properties measured for each neuron type. Information was available from experiments on a wide range of these properties: neuronal input resistance, resting potential, firing threshold, current threshold for firing, action potential (AP) height and width, maximum rise and fall rates of the action potential, after-hyperpolarisation amplitude and delay, initial firing frequency to long, constant current input, and the effect of increasing current on this frequency (see Table 5.3; [90], [92], [93], [96]).

The general equations are based on the Hodgkin-Huxley model, as described in Section 2.2.1. This was mainly because these equations allow monitoring of individual ionic currents, and because the model by Dale [24], used as starting point, was of that type. The differential equation for the voltage is:

$$C\frac{dV}{dt} = I_{inj} + I_{syn} - Currents\ , \tag{5.1}$$

where C represents the cell's membrane capacitance (based on measurements), and is set to 4 pF for all neurons. I_{inj} represents experimentally injected current and I_{syn} the ensemble of synaptic currents. $Currents$ is the combination of all modelled ion channel currents. The number of these depends on the cell type, but all neurons have at least the following currents:

$$\begin{aligned} Currents \quad = \quad & g_{leak}(V - V_{leak}) + g_{Na}m_{Na}^{3}h_{Na}(V - V_{Na}) \\ & + g_{Kfast}n_{Kfast}(V - V_{Kfast}) + g_{Kslow}n_{Kslow}(V - V_{Kslow}). \end{aligned} \tag{5.2}$$

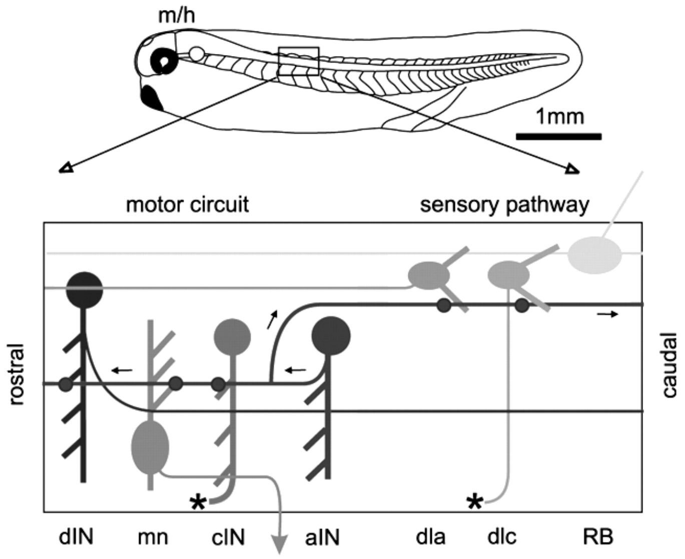


Figure 5.4: Schematic picture of the neurons and synapses in the *Xenopus* spinal cord. The diagram shows the different neuron types that are active during swimming, and their dorsoventral position. (Picture from [95].)

The currents include one leak current, one Na-current and two K-currents, one fast and one slow.

The leak conductance (g_{leak}) is the inverse of the measured input resistance (rescaled to nS), and the leak reversal potential (V_{leak}) is given by the resting potential of the cell. This is clearly a mixed current, since in all neurons it was less negative than the reversal potential for Potassium. The maximum conductances for Sodium and Potassium (g_{Na}, g_{Kfast} and g_{Kslow}) are based on Dale ([24], [25]) and matched to experimental results. The reversal potentials for Sodium and Potassium (V_{Na}, V_{Kfast} and V_{Kslow}) are based on physiological measurements. In the absence of detailed measures, the neuronal input capacitance was assumed to be the same for all neurons and was set to 4 pF.

The Na-current has both an activation (channel opening) and an inactivation (channel closing) component, these are represented by m and h. The K-currents only have the activating part (n_{Kfast} and n_{Kslow}), so are for present purposes assumed to be non-inactivating; cf [24]. The behaviour of these activations and inactivation components is defined by the equation:

$$\frac{dgate}{dt} = \frac{1}{\tau_{\_}}(max_{\_} - gate) \ , \tag{5.3}$$

where $gate$ is any of the above four variables, t is time, $\tau_{\_}$ is the time constant of the particular type of channel, and $max_{\_}$ is the maximum opening rate of this type of channel. $\tau_{\_}$ and $max_{\_}$ are voltage dependent, and thus are also defined by equations:

$$\begin{aligned} \tau_{\_} &= \frac{1}{\alpha_{\_} + \beta_{\_}} \\ max_{\_} &= \frac{\alpha_{\_}}{\alpha_{\_} + \beta_{\_}} \end{aligned} \tag{5.4}$$

where

$$\begin{aligned} \alpha_{\_} &= \frac{A_1 + B_1 V}{C_1 + \exp(\frac{D_1+V}{E_1})} \\ \beta_{\_} &= \frac{A_2 + B_2 V}{C_2 + \exp(\frac{D_2+V}{E_2})} \ . \end{aligned} \tag{5.5}$$

Here V is again the membrane potential, A_i, B_i, C_i, D_i and E_i are parameters to be set for each type of neuron and channel.

5.2.2 Type-specific models

Our objective was to build models of each of the known spinal neuron types, as listed in Tables 5.1 and 5.2 [115].

We have divided these neurons into four groups based on their very clearly different and characteristic firing responses to injected current. We then set parameter values in order to match each model to the physiological properties of each individual neuron type. In life, the properties of individual neurons within each type vary and without using population models we cannot model this variability. Our aim was to produce model neurons whose properties encapsulated the main physiological features of each neuron type. This would allow us to explore the significance of the different properties in the behaviour of the swimming circuit.

Unless stated otherwise, we used the generic equations described in Section 5.2.1. We decided that all neurons should have fast and slow K-channels as this seems more biologically realistic. Where Dale used discontinuous functions to represent the potassium currents, we used continuous functions instead. For simplicity we have not used any calcium currents.

The parameters were based on [25], but to get better spiking behaviour in the model neurons, and match other physiological data, we have made parameter changes to the data from [25].

For all types of neurons, the measured physiological parameters are listed in Table 5.3 and the model neuron parameters are listed in Tables 5.4, 5.5 and 5.6.

Input resistance and resting potential were matched by setting the leak conductance and leak reversal potential. To set the firing threshold, the D-parameters of all currents, and some of the A-parameters, needed to be changed from model to model. One method to do this [114] involves decreasing the spiking threshold by an amount X for all currents, as follows:

$$\begin{aligned} A' &= A + X\,B \\ D' &= D + X\,. \end{aligned} \tag{5.6}$$

Then A' and D' are the new values. In several model neurons, we had to modify the maximum conductances of the ion-channels to match recorded responses.

	aIN	**MN**	**dIN**	**RB**	**dlc**	**dla**	**cIN**
Input Resistance (MΩ)	740	405	272	230	428	1436	206
Resting potential (mV)	-54	-61	-51	-70	-66	-63	-60
Current threshold for firing (pA)	28	109	85		95	53	198
Action potential (AP) threshold (mV)	-29	-26	-28	-25	-35	-32	-25
AP peak (mV)	27	26	29	40	28	37	32
AP width (ms)	1.8	0.9	1.9	1.5	0.5	0.75	0.8
Max. AP rise rate (mV/ms)	121	105	85		127	130	124
Max. AP fall rate (mV/ms)	97	77	76		137	100	101
Initial firing frequency (Hz)	24	41			97	29	71
Slope of frequency-current relationship (Hz/pA)	53				67		84
Afterhyperpolari-zation (AHP) trough (mV)	-47	-42	-43		-44	-57	-43
AHP delay (ms)	1.8	2.3	6.9		1.3	2.1	1.5
Response to injected current	RF*	RF*	S*	S*	AF*	AF*	DF*
Initial gap before delayed burst (ms)							167
Slope of gap-current relationship (ms/pA)							-200

*RF = Repetitive Firing, S = Single spike
AF = Adapting Firing, DF = Delayed Firing

Table 5.3: Measured properties for the different neuron types based on whole-cell recordings (medians, based on 5 - 20 neurons of each type).

	aIN	MN	dIN	RB	dlc	dla	cIN
C	4.00	4.00	4.00	4.00	4.00	4.00	4.00
g_{Leak}	1.3514	2.4691	3.6765	4.3573	2.3364	0.6964	3.8544
V_{Leak}	-54.00	-61.00	-51.00	-70.00	-66.00	-63.00	-60
g_{Na}	150.00	110.00	210.00	120.00	420.00	150.00	500
V_{Na}	50.00	50.00	50.00	50.00	50.00	50.00	50
g_{Kfast}	15.00	8.00	0.50	1.50	70.00	70.00	30
V_{Kfast}	-80.00	-80.00	-80.00	-80.00	-80.00	-80.00	-80.00
g_{Kslow}	2.50	1.00	3.00	8.00	10.00	5.00	20
V_{Kslow}	-80.00	-80.00	-80.00	-80.00	-80.00	-80.00	-80.00
g_A							30
V_A							-80.00

Table 5.4: Parameters for all modelled neurons: capacitance (in pF), the conductances (in nS) and reversal potentials (mV) of their currents.

Repetitively firing neurons

aINs, dlxs, MNs and dINrs all fire repetitive action potentials to depolarising current injection but there are big differences in input resistance.

aINs and MNs are most like Dale's original neurons [25]. They fire repetitively to sustained injected current (Figure 5.5), but there are differences, for example in input resistance and firing frequency during current injection [90].

aIN

Starting from Dale's parameters [25], several changes were needed to fit our measurements. The D-parameters to all currents were decreased, to increase the action potential voltage threshold. To increase the spike frequency to depolarizing current, the time constants τ for the Na-activation and -inactivation were changed: τ for the activation (m_{Na}) was decreased by increasing the A-parameter of its β rate constant, and even more at high voltages by decreasing the C-parameter of the α rate constant; τ for the inactivation (h_{Na}) was increased by decreasing the A-parameter of its α rate constant, but restored at high voltages by decreasing the

Dale’s model	A	B	C	D	E
αm_{Na}	8.67	0	1	-1.01	-12.56
βm_{Na}	3.82	0	1	9.01	9.69
αh_{Na}	0.08	0	0	38.88	26
βh_{Na}	4.08	0	01	-5.09	-10.21
αN_{Kfast}	3.1	0	1	-29.5	-23.3
βN_{Kfast}	0.44	0	1	6.98	16.19
αN_{Kslow}	0.16	0	1	-4.96	-7.74
βN_{Kslow}	0.04	0	1	-16.07	6.1
aIN / MN	A	B	C	D	E
αm_{Na}	8.67 / 13.26	0	0.5	-13.01 / -5.01	-18.56 / -12.56
βm_{Na}	5.73	0	1	-2.99 / 5.01	9.69
αh_{Na}	0.04	0	0	15.8 / 28.8	26
βh_{Na}	4.08 / 2.04	0	0.001	-19.09 / -9.09	-10.21
αN_{Kfast}	3.1	0	1	-35.5 / -27.5	-9.3
βN_{Kfast}	1.1 / 0.44	0	1	0.98 / 8.98	16.19
αN_{Kslow}	0.2	0	1	-10.96 / -2.96	-7.74
βN_{Kslow}	0.05	0	1	-22.07 / -14.07	6.1
dIN / RB	A	B	C	D	E
αm_{Na}	13.01	0	4 / 1	-1.01 / -4.01	-12.56
βm_{Na}	5.73	0	1	9.01 / 6.01	9.69
αh_{Na}	0.06 / 0.04	0	0	30.88 / 29.88	26
βh_{Na}	3.06 / 2.04	0	1	-7.09 / -8.09	-10.21
αN_{Kfast}	3.1	0	1	-31.5 / -32.5	-9.3
βN_{Kfast}	0.44	0	1	4.98 / 3.98	16.19
αN_{Kslow}	0.2	0	1	-6.96 / -7.96	-7.74
βN_{Kslow}	0.05	0	2	-18.07/ -19.07	6.1

Table 5.5: (Part 1) Detailed parameters for the currents in the modelled neurons. The top table lists the original parameters from [25].

dlc / dla	A	B	C	D	E
αm_{Na}	13.26	0	3 / 1.2	-3.01 / -9.01	-12.56
βm_{Na}	5.73	0	1	6.01 / 1.01	9.69
αh_{Na}	0.06 / 0.04	0	0	19.88 / 14.88	26
βh_{Na}	4.08 / 2.04	0	0.001	-8.09 / -13.09	-10.21
αN_{Kfast}	3.1	0	1	-32.5 / -37.5	-9.3
βN_{Kfast}	1.1	0	2 / 0.6	3.98 / -1.02	16.19
αN_{Kslow}	4	0	1	-53 / -58	-7.74
βN_{Kslow}	0.01	0	1	47 / 42	6.1
cIN	A	B	C	D	E
αm_{Na}	13.26	0	0.1	-10.01	-12.56
βm_{Na}	5.73	0	1	0.01	9.69
αh_{Na}	0.06	0	0	23.8	26
βh_{Na}	3.06	0	0.001	-14.09	-10.21
αN_{Kfast}	3.1	0	1	-32.5	-9.3
βN_{Kfast}	1.1	0	1	3.98	16.19
αN_{Kslow}	0.2	0	1	-7.96	-7.74
βN_{Kslow}	0.05	0	0.5	19.07	6.1
αm_A	12.025	0	0.5	-10.01	-12.56
βm_A	14.325	0	1	-8.01	9.69
αh_A	0.0001	0	1	15.88	26
βh_A	10	0	500	-22.09	-10.21

Table 5.6: (Part 2) Detailed parameters for the currents in the modelled neurons.

C-parameter of its β rate constant. The D-parameters of the Na-current were decreased more than those of the K-currents, to lower the threshold current and time constant further. To lower the time constant of Na-activation around threshold and zero voltage (i.e. to get shorter spikes), its E-parameter was decreased, and to increase the time constant of the fast K-activation in that range, its E-parameter was increased. The maximum activation of the fast K-channels was increased, and its time constant decreased, by increasing the A-parameter of its β rate constant. The time constant of the slow K-current was decreased, but it plays a less important role, as its maximum conductance is relatively low.

MN

These neurons are very similar to aINs. To match the threshold behaviour (with the changed resting potential), the D-parameters of all currents were changed, and the activation was sped up by increasing the A-parameter of the α rate constant of the Na-activation. All maximum conductances were decreased, to match threshold current and voltage. The E-parameter of the α rate constant was increased again to the value used by Dale [25]. The A-parameter of the β rate constant of K$_{fast}$ was decreased to get a deeper AHP and the A-parameter of the β rate constant of the Na-inactivation was increased. To reduce spike width, the D-parameters of the Na-inactivation were increased more than others, which results in a lower time constant.

Single-spiking neurons

dINs and RBs fire a single, long-duration action potential at the start of current injections and only rarely give any further response (Figure 5.6; [91], [96]).

dIN

A unique property of dINs is that they never fire repetitively to sustained current, no matter how strong it is. To match this, starting from the aIN parameters, we increased the C-parameters of the α rate constant of Na-activation (m_{Na}; from 1 to 4), of the β rate constants of Na-inactivation (h_{Na}), and of the slow K-current. In addition, the maximum conductance of slow K was increased, and that of fast K was decreased. To match the change in firing threshold, and to compensate

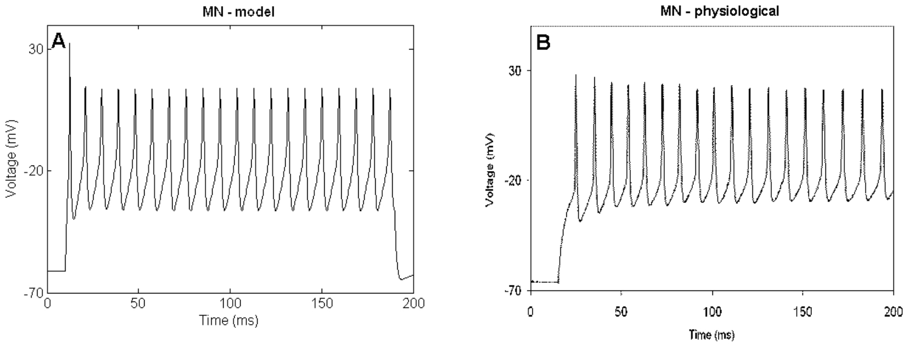


Figure 5.5: Responses of repetitively firing neuron model. **A**: Output of model MN firing in response to injected current. **B**: Physiological measurement of repetitively firing neuron (MN).

for the big difference in input resistance (compared to aINs), all D-parameters were increased. To match the threshold, the D-parameters of the Na-currents were changed more than the other D-parameters. The parameters for the β rate constant of the (less important) fast K-current were set to Dale’s values. To match spike width and AHP, the A-parameters of the Na-inactivation were changed, which causes a larger time constant and a higher maximum value (i.e. wider spikes and larger impact).

A property of dINs, which turns out to be crucial in network behaviour is that they do not fire on rebound when negative current pulses are given at rest but have the ability to fire on rebound if a negative current pulse of sufficient amplitude is given during depolarisation (see Fig 4E in [96]). This property is also present in our dIN model (Figure 5.7). It is not known if this property is exclusive to dINs.

RB

RB neurons have a rather negative resting potential. Starting from the dIN current parameters, most of the D-parameters were changed (except for the Na-activation) to get the right threshold voltage and current. The time constant of the sodium inactivation (h_{Na}) was reduced, to reduce spike width, by decreasing

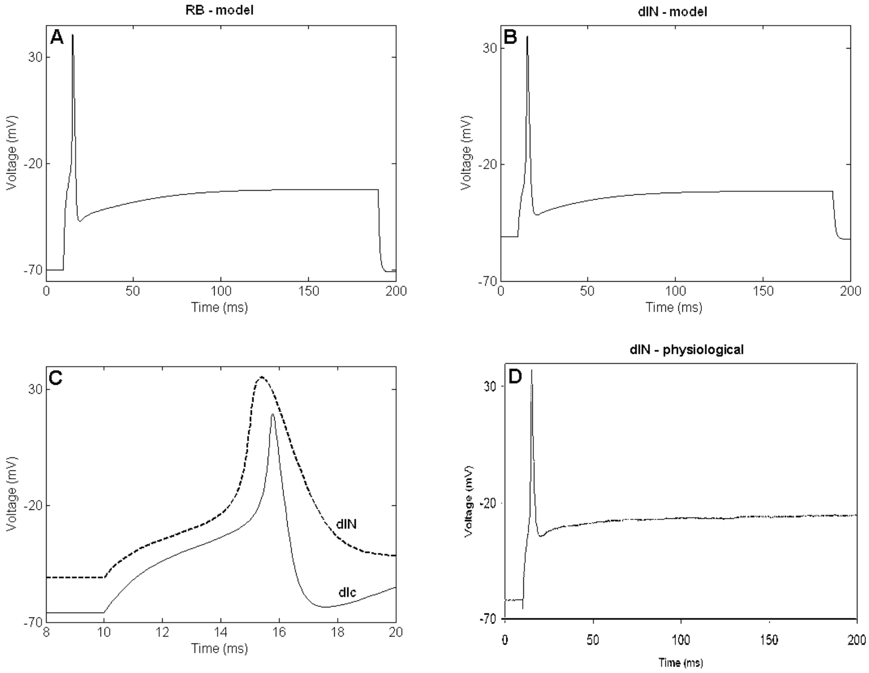


Figure 5.6: Responses of single-spiking neuron models to injected current. **A**: RB. **B**: dIN. **C**: Wide dIN spike (thick dashed line) superimposed on then dlc spike (thin solid line). **D**: Physiological measurement from single-spiking neuron (dIN).

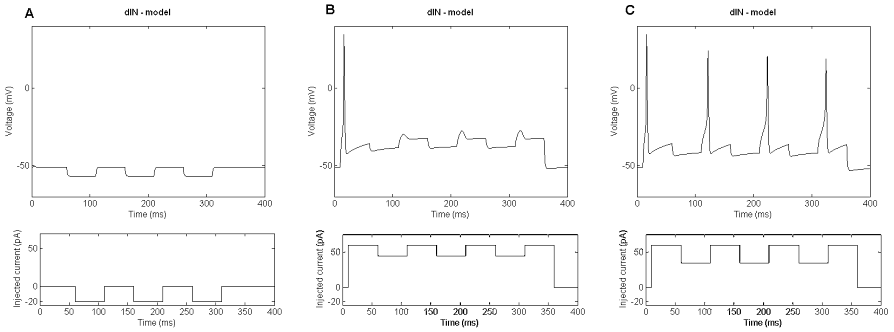


Figure 5.7: Rebound firing in dINs. **A**: dINs do not fire rebound spikes when negative current is given at resting potential. **B**, **C**: They do fire on rebound after a negative current pulse during depolarisation, if the pulse is of sufficient amplitude.

its A-parameters. The maximum conductance of the slow K-current could be increased to suppress repetitive firing, and that of the Na-current decreased, which allowed the C-parameters of the α rate constant of Na-activation (m_{Na}) to be reduced to 1, as in Dale’s model.

Repetitively firing neurons with adaptation

When current is injected into sensory pathway interneurons (dlc and dla), they fire repetitively, but the frequency drops and firing stops quickly (Figure 5.8; [120], [92]). To model this, the slow K-current was modified.

dlc

To get the adapting behaviour, we wanted the slow K-current to increase at each spike fired, and remain sufficiently activated so that the neuron would not start to fire again after a while. To obtain this, the D-parameters of the slow K-current were decreased for the α rate constant and increased for the β rate constant. The A-parameters of these rate constants were increased and decreased respectively. The other D-parameters were decreased (compared to aIN) to match the threshold voltage and current. By increasing the C-parameter of the α rate constant of m_{Na}, the threshold for repetitive firing was raised. In this way, once repetitive fir-

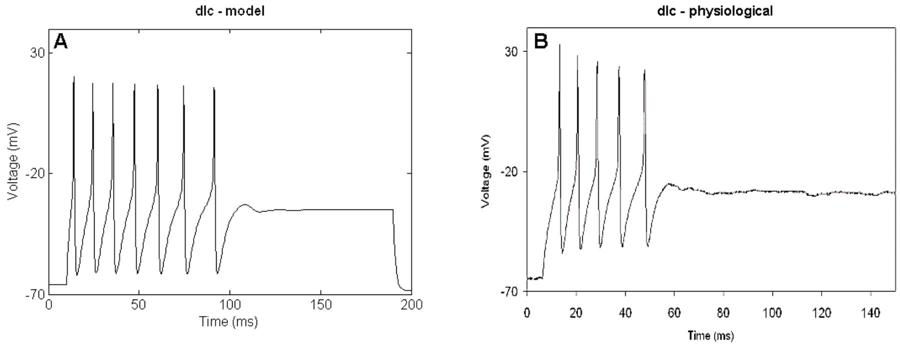


Figure 5.8: Responses of repetitively firing neuron model with adaptation. **A**: Output of model dlc firing in response to injected current. **B**: Physiological measurement of repetitively firing neuron with adaptation (dlc).

ing starts, the frequency is higher. And it is specifically this high initial frequency that was needed. To the same effect, the D-parameters of the fast K-current were decreased (more than the other current’s D-parameters), and the maximum conductances of Na- and fast K-currents were increased.

dla

To compensate for the much higher input resistance and lower threshold current, the conductances of all currents were decreased (starting from dlc parameter values). To then match the threshold voltage and current, all D-parameters were decreased slightly, and the one for the α rate constant of the Na-activation more than the others. To increase spike width, the A-parameters for Na-activation were decreased, and to get a low initial firing frequency, the C-parameters of the α rate constant of the Na-activation and of the β rate constant of the fast K-current were decreased.

Delayed bursting neurons

Neurons of one specific type (cINs) exhibit a delayed burst in response to constant input current ([1]; see Figure 4B in [94]). Typically, this burst follows a delay which is preceded by a single, initial spike (Figure 5.9). To model this property, a further K-current was used, that has both an activation and an inactivation component [51]. A current of this type has previously been described for *Xenopus* neurons developing in culture [108]. This current results in an extra term added to equation (5.2).

$$\begin{aligned} Currents &= g_{leak}(V - V_{leak}) + g_{Na}m_{Na}^3 h_{Na}(V - V_{Na}) \\ &\quad + g_{Kfast}n_{Kfast}(V - V_{Kfast}) + g_{Kslow}n_{Kslow}(V - V_{Kslow}) \\ &\quad + g_A m_A^3 h_A(V - V_A) \end{aligned} \tag{5.7}$$

cIN:

The characteristic firing property of cINs is repetitive firing following with a delay after an initial single spike. The maximum conductance for the Na-current was increased significantly (relative to the other currents) to overcome the higher leak conductance and get a high firing frequency. All current parameters (A, B, C, D and E) were kept similar to those for aINs. The maximum conductances of all currents were altered, to obtain the right firing threshold. The C-parameter for the α rate constant of the Na-activation was decreased, which lowers the time constant for high voltage, resulting in narrow spikes and fast repetitive firing. To match spike width further, small changes were made to the A-parameters of the Na-inactivation. To increase the time constant of the fast K-current, the C-parameter of its β rate constant was decreased.

The new K-current, added to produce the delay in repetitive firing, was initially given the same parameters as the Na-current, which also shows both activation and inactivation. The D-parameters were then decreased to match threshold and spiking behaviour, but the α rate constant for activation of the current (m_A) was unaltered to maintain the pre-burst delay. The C-parameter for the inactivation rate constant β was increased to 500: the higher this value, the longer the delay. The inactivation of the new current should hardly change when there is no injected current. To obtain this, the C-parameter of its α rate constant is set to 1, and the A-parameter very low. The A-parameter of its β rate constant is set to 10, to decrease the maximum opening rate. The A-parameters of the activation of the new current are high, so that it is fast enough to suppress immediate repetitive firing after the initial spike in response to sustained current. The slow inactivation

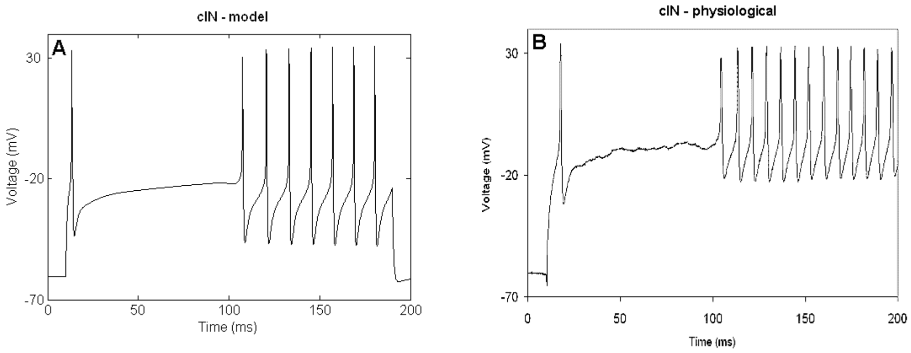


Figure 5.9: Responses of delayed firing neuron model. **A**: Output of model cIN firing in response to injected current. **B**: Physiological measurement of delayed firing neuron (cIN).

of the new current will allow repetitive firing after a delay.

5.3 Synapse models

In a biologically realistic neural network, you not only need realistic models for the individual neurons, but also realistic connections between these neurons. That way the communication over the network can be regulated in a physiologically relevant way. Therefore, we have developed relatively accurate models for the synapses in the *Xenopus* tadpole.

In our network, we use 3 different types of chemical synapses: glutamatergic AMPAR, and NMDAR mediated excitatory synapses, and glycinergic inhibitory synapses. The NMDAR mediated excitation is voltage dependent ([122], [114], [25]). Although dINs are known to co-release acetylcholine and glutamate we have not included a separate acetylcholine component. The time course of the acetylcholine current is similar to the AMPA component so we have combined the two as one single 'fast' component [93].

5.3.1 Equations

The time evolution of each type of synapse is implemented as the subtraction of 2 exponential functions, the opening function o and the closing function c. These functions are increased stepwise: every time a presynaptic spike is fired (a maximum voltage is reached above zero mV), the synaptic functions are increased with the fixed value of 10. Then the functions decay exponentially. In equation (5.1), I_{syn} stands for the sum of all synaptic inputs. This synaptic signal is, for a signal from neuron i to neuron j, defined as:

$$I_{syn}(i,j) = Gmx\,(c-o)\,(Er-V_j)\,CS(i,j)\;. \tag{5.8}$$

Here Gmx and Er are the maximum conductance (in nS) and the reversal potential (in mV) of that type of synapse, respectively. V_j is the membrane potential (in mV) of postsynaptic neuron, o and c are the dimensionless opening and closing functions of the synapse, and $CS(i,j)$ is the dimensionless connection strength of this particular synapse. The decay of the opening and closing function is described by the following exponential functions:

$$\begin{aligned} \frac{dc}{dt} &= -\frac{1}{\tau_c}c \\ \frac{do}{dt} &= -\frac{1}{\tau_o}o \end{aligned} \tag{5.9}$$

where the τ_o and τ_c, the opening and closing time constants, are parameters for each type of synapse.

5.3.2 Synapse parameters

Table 5.7 lists the parameters used in the synaptic equations described above, together with the references we got the values from. Modifications (not in italics) in the values for τ_o and τ_c of the inhibitory and AMPA synapses were made to fit the experimental data, with the results of this fitting shown in Figures 5.10 and 5.11. All inhibitory synapses are the same but in excitatory synapses the ratios of AMPA to NMDA conductances varied (Figure 5.10B, C).

	Inhibitory	**AMPA**	**NMDA**	**References**
Gmx (nS)	*0.435*	*0.593*	*0.29*	[50],[55],[16]
τ_o (ms)	1.5	0.2	*5*	[24]
τ_c (ms)	4	3	*80*	[24]
Er (mV)	-52 (dIN -64)	0	*0*	[114], unpub. measurements

Table 5.7: Parameters used for chemical synapses. The parameters were based on the references listed, but the time constants of inhibitory and AMPA synapses were modified slightly to fit the experimental data. The unchanged parameters are in italics.

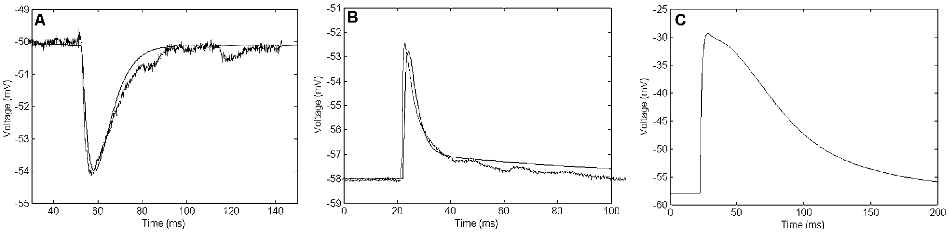


Figure 5.10: Post-synaptic potentials at synapses. Lighter straight line = response of a leak-only neuron model (no ionic currents involved), Darker jittery line = physiological measurements. **A**: Inhibitory postsynaptic potential. **B**: Fast excitatory postsynaptic potential with 50% AMPA and 50% NMDA. This ratio is comparable to e.g. the RB to dlc synapse. **C**: Slow excitatory postsynaptic potential with 35% AMPA and 65% NMDA. This ratio is comparable to the dIN feedback synapse.

5.3.3 NMDA voltage dependence

Voltage dependence of the NMDA synapses is mediated by Mg^{2+} ions outside the neuron which at resting membrane potential block the NMDA receptor controlled channels, until the postsynaptic neuron is sufficiently depolarized to eject the Mg^{2+} ions [103]. The voltage dependence of the NMDA synaptic current is added by slightly modifying (5.8) for the NMDA synapses:

$$I_{syn}(i,j) = Gmx\,(c-o)\,(Er-V_j)\,CS(i,j)\,X\ , \tag{5.10}$$

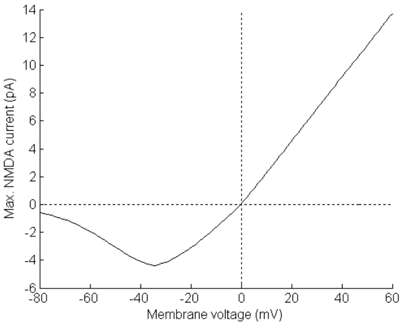


Figure 5.11: NMDA voltage dependence. The size of the peak NMDA synaptic current plotted against the membrane potential of a postsynaptic cell modelled with only leak conductances. This plot shows the progressive reduction in synaptic current below -35 mV reflecting the ’negative slope conductance’ produced by widely reported Mg^{2+} block of the NMDA channel.

where X is the voltage dependence factor, defined as:

$$X = (1 + 0.1 \text{ Mg } \exp(-0.08\, V_j))^{-1} \,. \tag{5.11}$$

This implementation is based on Lisman et al. [97]. Here Mg is the extracellular Mg^{2+} concentration in the neuron, which is approximated by the constant 0.5 mM. The voltage dependence of the NMDA current is shown in Figure 5.11 [122].

Chapter 6

Network models for the young *Xenopus* tadpole

A complex system that works is invariably found to have evolved from a simple system that works. (John Gaule)

In this chapter, we will use the individual neuron and synapse models from the previous chapter in biologically realistic networks. We have developed two different networks so far. The first network is a network to simulate the swimming behaviour of the *Xenopus* tadpole. Besides generating reliable swimming, an important aim for this model is to investigate the significance of type-specific properties of neurons in the operation of neuronal networks, using the spinal neurons and networks responsible for generating swimming activity in young tadpoles. Then we build a network model to test a hypothesis on synapse generation. For this, we build a network where whether different neurons connect through synapses depends purely on statistics and probabilities. The idea is that this is what happens during synaptogenesis (= generation of synapses), and that synapses have no special neuron recognition mechanism.

6.1 Types of behaviour

Like the different locomotor gaits shown by mammals, the neuronal network in the young tadpole spinal cord is able to generate two rather different patterns of motor output. In response to a brief touch to the skin, the tadpole swims using alternating lateral bends in the trunk that move in a wave from head to tail at frequencies from 10 to 25 Hz [81]. When tadpoles are held and the skin stimulated continuously they struggle making stronger, slower bends at 2 to 5 Hz that move from the tail to the head [80].

In immobilized tadpoles, very different patterns of activity in spinal interneurons and motoneurons can be recorded during these two responses. During swimming spinal neurons fire a single action potential on each cycle, but during struggling they fire high frequency bursts lasting nearly half a cycle [123]. Recordings have shown that motoneurons and some inhibitory interneurons are active during both activities [123]. During swimming, if driven by brief excitation from dINs, these neurons will only fire single spikes, but during struggling, with slower, more sustained excitation, they have the capacity to fire fast and repetitively in bursts. Other repetitively firing interneurons, the inhibitory aINs, usually fire a single spike on each swimming cycle but can sometimes fire twice and are vigorously active during struggling [90]. Finally, some repetitively firing interneurons that are not active during swimming become active and fire bursts during repetitive skin stimulation that leads to struggling. The role of repetitively firing neurons in the struggling network is the subject of current physiological and modelling studies (Li et al., in preparation).

So far, we have primarily focussed on the swimming behaviour. To model the tadpole's struggling is one of our future goals.

6.2 The *Xenopus* swimming network

6.2.1 dIN rebound properties in small networks

Experimental work has given direct evidence that the excitatory dINs play a key role in driving the other neurons that are active during the swimming rhythm [96]. However, since these neurons fire only a single spike when depolarised, it is important to consider how they can be made to fire further spikes before exploring the operation of the whole swimming circuit.

As mentioned in Section 5.2.2, dINs can fire spikes on rebound following negative current injection, provided they are depolarised [96], and it has been proposed that this property may have a key role in rhythm generation. Neurons depolarized by synaptic excitation could fire on rebound after reciprocal inhibition from neurons on the opposite side of the spinal cord [114].

We therefore tested if the model dIN would fire on rebound following synaptic inhibition. Consider a dIN receiving a fast excitatory synaptic input, followed by an inhibitory one. If the IPSP occurs near resting potential, it will not cause any rebound spike (Figure 6.1**A**). However, when slow feedback excitation depolarizes the dIN after external stimulation, it will still only fire a single spike after the fast excitatory synaptic input (Figure 6.1**B**), but it will fire a rebound spike after the IPSP (Figure 6.1**C**). This is precisely the mechanism that supports the continuation of activity during swimming.

6.2.2 Swimming network

Activity of spinal neurons - network diagram

In our model of the circuitry for swimming and its sensory initiation, single neurons of each type are used to represent what in the real animal are populations of similar neurons. These populations typically consist of several hundreds of neurons throughout the spinal cord. The network architecture is based on the anatomical evidence on axonal projections and the synaptic connections are based on results from paired recordings. The RB neurons are touch sensory neurons innervating the skin. Thus, when the *Xenopus* skin is touched, the RB neurons fire [18]. The RB neurons excite sensory interneurons (dlc and dla), which relay the signals through excitatory synapses to the contralateral (opposite) and ipsilateral (same) side, respectively [91], [92]. Here they synapse onto all ’central pattern generator’ (CPG) neurons, including dINs [91], which in turn excite all CPG neurons on their own side [96]. These other CPG neurons are aINs, which provide ipsilateral, recurrent inhibition (i.e. returning inhibition on the same side) [95],

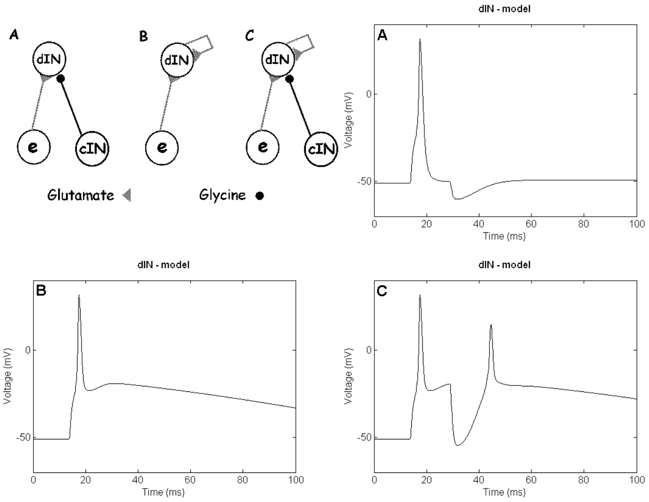


Figure 6.1: Post-inhibitory rebound in dINs. Top left: Diagrams showing small networks used to create plots **A**,**B** and **C**. **A**: dIN response to fast excitatory input (main component is AMPA from neuron e), followed by inhibitory input. **B**: dIN response to fast excitatory input, followed by slow feedback excitation (main component is NMDA). **C**: dIN response to fast excitatory input, followed by slow feedback excitation and then inhibition, causing a rebound spike.

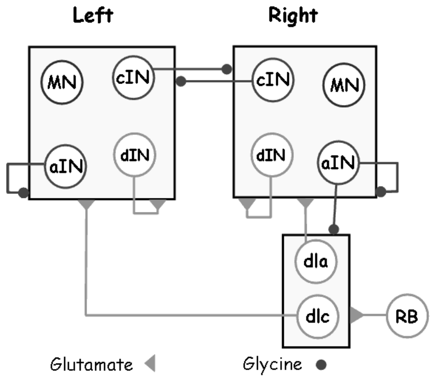


Figure 6.2: Diagram of the neuronal network which generates swimming. Triangles stand for excitatory synapses, circles for inhibitory synapses. A synapse to a ’box’ means the connection is made to all neurons inside the box.

cINs, which provide contralateral, reciprocal inhibition (i.e. inhibition to other neurons on the other side of the body) [22], [124] and MNs, which connect to the muscles and cause the swimming movements [121].

The model network (Figure 6.2) has a left and right side but no length and represents the neurons and connections found in the caudal hindbrain and rostral spinal cord. All ipsilateral synapses were assumed to have a combined axonal-synaptic delay of 1 ms, all commissural synapses (i.e. passing from one side of the spinal cord to the other) were given a delay of 2 ms as they are made from axons which have crossed from the other side of the spinal cord.

The modelling work itself is done in MATLAB, with models written in that language and simulations run using MATLAB’s built-in ODE suite. The individual models for neurons and synapses are written as ordinary differential equations, but with discontinuities incorporated in the network behaviour: there are discrete increments in synapse parameters at times of a presynaptic spike, and the delays are also part of the network behaviour, rather than being included in the differential equations. The total number of equations comes to 52 for the neurons (6 equations for each cIN and 5 equations for all other types), and 68 for the synapses (2 equations per synapse).

Synaptic conductances

During swimming, some of the maximum conductances were measured directly in physiological experiments (Table 6.1, on the left). Of those that were not (or could not be) directly measured, estimates were made (for example, using AMPA:NMDA ratios measured from unitary synaptic potentials). The (dimensionless) synaptic strengths in the model swimming network were set to specific values, as to produce conductances whose peak values during swimming (Table 6.1, on the right) approximate the measured (or estimated) values. These values, measured (or estimated) for a particular type of synaptic connection, were actually the sum of synchronous inputs from a population of neurons of the same type. Since populations of neurons are represented as individual neurons in the model, each connection type was represented as a single synapse whose conductance, as in a real neuron, was equivalent to that produced by input from a population of neurons.

Swimming result

Unlike previous models of the spinal cord, which lacked sensory pathway neurons (e.g. [111], [25]), a single 'stimulus' to one RB neuron (on the right side), leading to a single RB action potential, makes the network enter a swimming mode, where left and right sides fire alternating spikes. Exactly as in physiological experiments, the side opposite the stimulus (left) is active first and the firing rate is about 20 Hz (Figure 6.3**A**). As well as producing a clear swimming rhythm, the activity patterns of different neuron types resemble those seen physiologically, in a number of important respects (as shown in Figure 6.3**B**). Activity in the dINs is clearly different to that in the other CPG neurons, with a stronger background (tonic) depolarisation and more prominent reciprocal, mid-cycle IPSPs, followed on each cycle by rebound firing [96]. In contrast: the background depolarisation in the cINs, aINs and MNs is much smaller, and the mid cycle inhibition is hardly visible as the membrane potential is close to the inhibitory reversal potential [90], [96]. In these neurons the depolarisation is not sufficient for any to fire repetitively (as they are capable of doing to sustained current), or to fire on rebound. Instead,

		measured			**model**	
source	**target**	**AMPA**	**NMDA**	**A:N ratio**	**AMPA**	**NMDA**
RB	**dlc**	5.2^{1}	0.17^{2}	30.0^{3}	6.77	0.16
	dla	5.2^{1}	0.17^{4}	30.0^{4}	3.38	0.16
dlc	**dIN**	1.3^{5}	0.9^{6}	1.5^{3}	2.92	0.96
dla	**dIN**	1.3^{7}	0.9^{7}	1.5^{7}	2.55	1.08
dIN	**aIN**	1.6^{8}	0.5^{9}	3.3^{A}	1.46	0.54
	cIN	3.6^{8}	1.1^{9}	3.3^{A}	8.51	1.05
	dIN	4.2^{8}	4.2^{9}	1.0^{A}	4.38	6.86
	MN	5.1^{8}	1.5^{9}	3.3^{A}	3.89	1.47
		inhibition			**model**	
cIN	**aIN**	2.9^{B}			2.86	
	cIN	2.9^{B}			2.86	
	dIN	8.1^{B}			9.53	
	MN	3.9^{B}			3.81	
aIN	**cIN**	1.0^{C}			1.11	
	dIN	2.7^{C}			2.70	
	MN	1.3^{C}			1.27	
	dlc	5.0^{B}			6.36	
	dla	5.0^{B}			6.36	

Table 6.1: Physiological maximum conductances (in nS) for synaptic connections based on direct voltage clamp measurements and estimates, and maximum conductances reached in the model during swimming simulation. 1. Calculated from peak current in response to skin stimulation and assuming a reversal potential of 0 mV. - 2. Calculated using AMPA conductance and published AMPA/NMDA ratio [91]. - 3. From [91]. - 4. Calculated from measured AMPA conductance, assuming the same AMPA/NMDA ratio as for RB-dlc. - 5. Calculated from published peak current [91] in response to contralateral skin stimulation. - 6. Calculated from measured AMPA conductance and published AMPA/NMDA ratio [91]. - 7. Assumed to be the same as dlc-dIN but unconfirmed. - 8. Peak AMPA currents measured during swimming; conductance calculated assuming a reversal potential of 0mV. For these connections, AMPA is actually total ”fast” excitation (AMPA + nACh co-release; [93]). - 9. Calculated from measured AMPA conductance and published AMPA/NMDA ratio [93]. - A. Recalculated from data used in [93]. - B. Calculated from maximum peak currents and reversal potentials measured at mid-cycle during swimming. - C. Estimated using cIN values and assuming a cIN/aIN ratio of 3.

it is the relatively prominent fast excitation from the dINs that drives their single spike on each cycle. As a result, firing in the dINs leads firing in the other neurons on the same side (Figure 6.3**C**). The relatively weak excitation of aINs means that they fire relatively late on each cycle, as described experimentally [90].

If the level of excitation from dINs was reduced, similar swimming activity was seen but, as in 'spinalised' animals where excitation is similarly thought to be reduced [111], activity fails after a few cycles (Figure 6.3**D**).

To test the stability of the network, we varied the amplitude of the NMDA-component of the dIN feedback connection and of the cIN-dIN inhibition independently to establish the range over which swimming occurs (Figure 6.4). These synapses were chosen because they together control the rebound firing which drives the swimming network. In the gray area (both lighter and darker), there is stable swimming: the MNs give alternating output in the normal frequency range for swimming (15 to 25 Hz). Broadly: as the strength of dIN feedback excitation is increased from its lowest value, the frequency and number of swimming cycles increases until it becomes self sustaining; increasing the cIN-dIN inhibition decreases the frequency.

The strengths of other synapses can also play important roles in the network, but these are not as crucial to the production of alternating firing as the rebound firing mechanism. For example in Figure 6.5**A**, a frequency plot is shown for the swimming network when varying the AMPA and NMDA components of dIN feedback excitation. While variation in the NMDA strength clearly influences swimming frequency, variation of the AMPA strength has much less effect. Figure 6.5**B** similarly shows that varying cIN-dIN inhibitory conductance results in significant changes in swimming output, whereas varying cIN-cIN inhibition doesn't have any effect on swimming frequency.

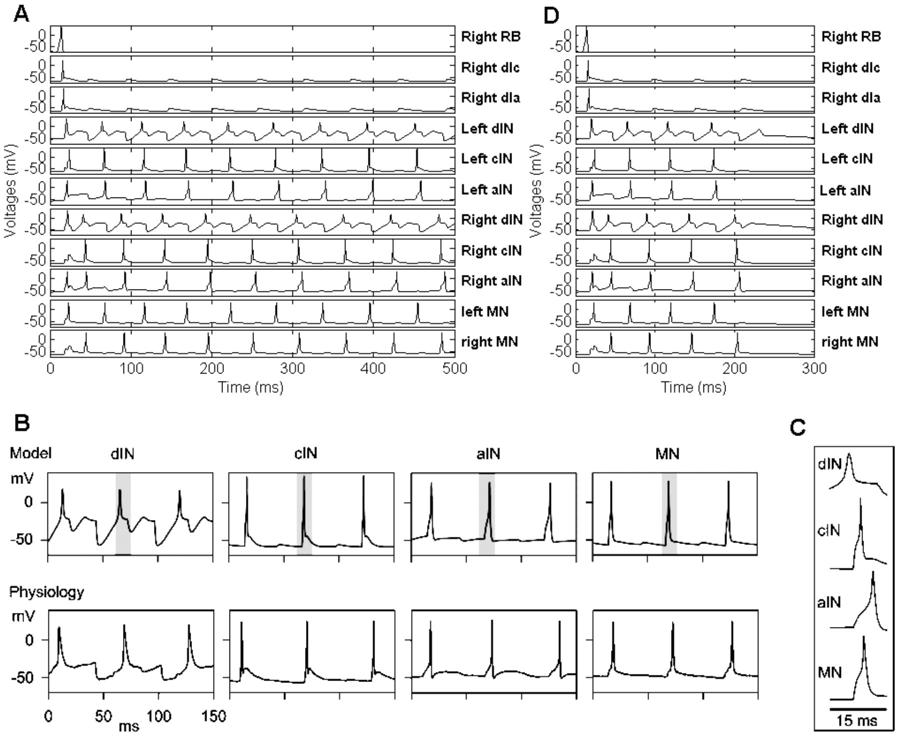


Figure 6.3: Swimming activity **A**: Voltage traces of all neurons in the network during sustained swimming, produced using the parameter values in Tables 2-5. **B**: Comparison of model output (sequences expanded from **A**) to physiologically recorded swimming activity of different CPG neuron types. **C**: dIN firing leads cIN and MN firing on each swimming cycle and aIN firing is relatively late. Data expanded from **B** (grey bars). **D**: With the strength of dIN feedback excitation reduced, swimming is sustained for only 4-5 cycles.

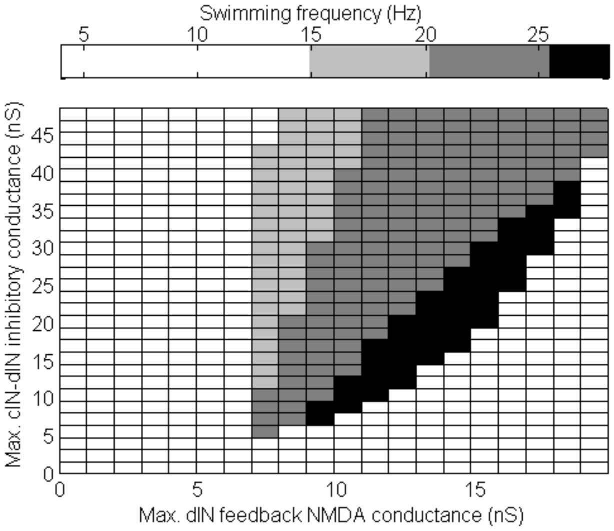


Figure 6.4: Surface plot of network swimming frequency, for different maximum conductance values of dIN feedback NMDA excitation and cIN-dIN inhibition. The gray area (both lighter and darker) has reliable swimming at a realistic frequency (15 to 25 Hz) and the black area has high frequency output, with synchrony between the two sides.

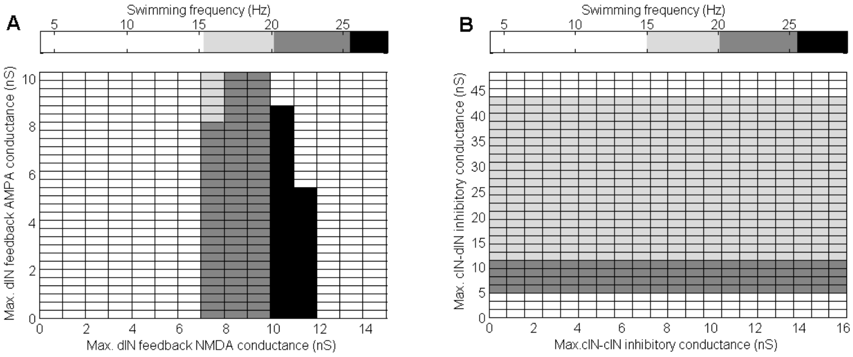


Figure 6.5: Influence of dIN feedback NMDA excitation and cIN-dIN inhibition on swimming. **A**: Surface plot of swimming frequency, for different maximum conductance values of dIN feedback NMDA and AMPA excitation. **B**: Surface plot of swimming frequency, for different maximum conductance values of cIN-cIN and cIN-dIN inhibition. Note: cIN-cIN inhibition has a smaller range, since its physiological conductance strength is much lower than the cIN-dIN inhibition.

6.2.3 Testing the importance of type-specific neuron models

Importance of single-spiking property of dINs

To investigate the importance of type-specific individual neuron models, we have focussed on the dINs. They provide excitation to drive the firing of the other active neurons on the same side, provide feedback excitation within each side, and receive reciprocal inhibition from contralateral cINs. These connections are crucial to allow the dINs to fire on rebound and therefore to drive swimming. Having established the stable area of swimming with dINs in the network (Figure 6.4), we changed the firing properties of the dINs.

We first gave them the repetitive firing properties of aINs. This led to unstable activity. The frequency plot (Figure 6.6) shows that the area of stable swimming (2 levels of gray) is partly enclosed by an area of high frequency with synchrony between the two sides (black). Comparing this to Figure 6.4, shows that replacing the dINs by aINs reduces the area of stable swimming, and causes a large area of synchronous firing on both sides of the body which would not lead to meaningful

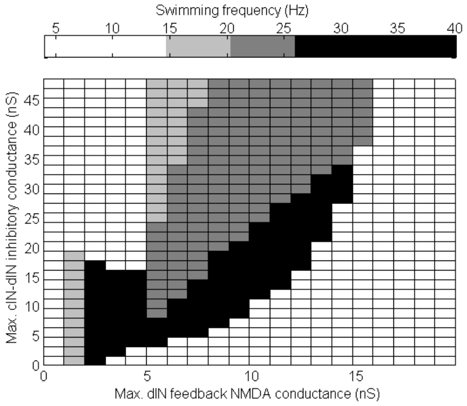


Figure 6.6: Frequency plot where dINs have the repetitive firing properties of aINs. The gray area (both lighter and darker) has realistic swimming output (15 to 25 Hz) and the black area has high frequency output, with synchrony between the two sides.

behaviour (see Figure 6.8).

When dINs were given the properties of cINs, they were unable to fire on rebound unless unrealistically high conductances were used, and in those cases, only synchrony or unstable firing patterns were produced. Substituting the properties of dINs suggests that their type-specific ”single-spiking” properties are important for the generation of stable swimming in this network model.

Importance of different neuron types

Since previous model networks have used a single generic neuron model [114], [26], [96], we made a network where all neurons had the same properties. In this case, our generic neuron model had the single-spiking property of dINs, rather than showing repetitive firing (as in [24]), since our modelling already suggested this was a key feature. Figure 6.7 shows a reliability surface for this network. There is swimming activity in about the same parameter region as in the regular swimming network (Figure 6.4).

This is perhaps not surprising, given that all CPG neurons typically fire only

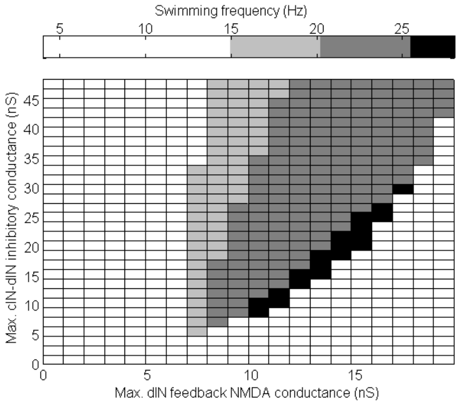


Figure 6.7: Surface plot of swimming frequency with all neurons having dIN properties. The gray area (both lighter and darker) has stable swimming at a realistic frequency (15 to 25 Hz).

once per cycle during swimming. However, it does raise the question of why most of the CPG neurons can fire repetitively. A possible explanation is that the spinal network must also generate the struggling pattern of motor output where neurons fire a high frequency burst on each cycle of a slower rhythm. Recordings have shown that motoneurons and some inhibitory interneurons are active during both activities [123]. During swimming, if driven by brief excitation from dINs, these neurons will only fire single spikes, but during struggling, with slower, more sustained excitation, they have the capacity to fire fast and repetitively in bursts. And some repetitively firing interneurons that are not active during swimming become active and fire bursts during repetitive skin stimulation that leads to struggling (dINrs and dlxs). The differences in firing properties may partly be a consequence of the need to switch rapidly between struggling and swimming, which may not make modulation of firing properties a practical alternative [62].

Importance of aINs in the swimming network

The key neurons in the swimming network appear to be the dINs providing excitation and the cINs providing reciprocal inhibition. We know that, in addition,

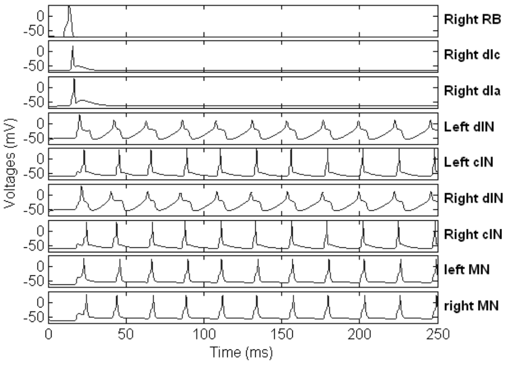


Figure 6.8: Example network output when aINs are removed. The network exhibits synchronous firing on both sides of the body, instead of alternating spikes.

aINs provide recurrent inhibition to CPG neurons [95], and this inhibition is particularly prominent as swimming is initiated.

When aINs are removed from the model network by setting their synaptic conductances to zero, the initiation of swimming always fails, leading only to high-frequency synchronous firing of MNs on both sides of the body. An example output is shown in Figure 6.8. This is the case even after varying several other synaptic conductances: the dlc/dla-dIN AMPA connection, dIN-dIN NMDA connection and cIN-dIN inhibitory connection were all varied over wide ranges, without producing good swimming behaviour. We can explain this role of aINs as follows: after stimulation of the right sensory RB neuron in our model, the right dla and dIN both excite the right aIN which fires quite quickly, and prevents the right cIN and MN from firing. Thus only the left cIN and MN fire, leading to rebound spiking in the right dIN, and swimming is started. When the aINs are removed, this process, that ensures swimming starts on only one side, is lost.

By increasing the dIN-aIN synaptic connection strength significantly (by a factor of 4), one can stop the network swimming: both aINs fire much sooner in the cycle, before the cINs do, and suppress the cIN spike. Thus any reciprocal inhibition is stopped, and the opposite dIN will receive no inhibition from which to fire on rebound. A similar effect is seen when the strength of aIN-CPG inhibition is increased sufficiently.

6.2.4 Role of aINs in the sensory pathway

In addition to their connections to other pattern generator neurons, aINs play a role in controlling sensory pathways. During swimming, sensory input from the skin via RB neurons is inhibited at certain phases of the swim cycle so that it does not lead to responses that are in conflict with swimming [119]. We now know that this gating inhibition of dlc and dla interneurons comes from aINs [90]. Since these connections are a part of our model network, sensory gating can operate. If the RB neuron gets an extra stimulus during swimming, the timing of the stimulus in the swimming cycle determines whether the dlas and dlcs will fire (Figure 6.9). Only if they fire will their excitation reach the CPG and lead to an increase in swimming frequency (Figure 6.10).

6.3 Probabilistic network

6.3.1 Aim of this network

In the previous section, we stated that each type of neuron connects to specific types, and does not connect to other types. Although in reality this is not a strict rule (connections can be found between any two types of neuron), it is clear that certain neurons appear to prefer some postsynaptic targets more than others. In this section, we want to study why this is the case.

When a neuron develops, the axon grows, and makes connections to other neurons. One possible cause for the type-specificity of the connections is that the growing axons would have a means to make a distinction between different types of neuron, and have the ability to decide whether the neuron reached is of an appropriate type to connect to [17].

However, another option is that whether an axon connects to some specific neuron is a matter of chance. If there were not any specific connectivity rules re-

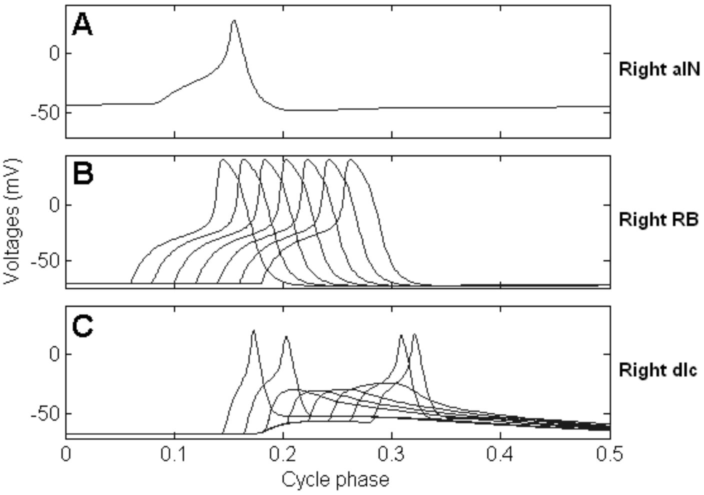


Figure 6.9: Gating of sensory transmission from RB to dlc. Experiments are done in the complete network model, but only the output of the three most important neurons is shown. **A**: aIN spike near the start of a cycle of swimming (phase 0). **B**: Series of spikes in RB from separate stimuli given at different times relative to the aIN spike. **C**: Depending on the relative timing of aIN and RB spikes, dlc firing can be suppressed by the aIN inhibition.

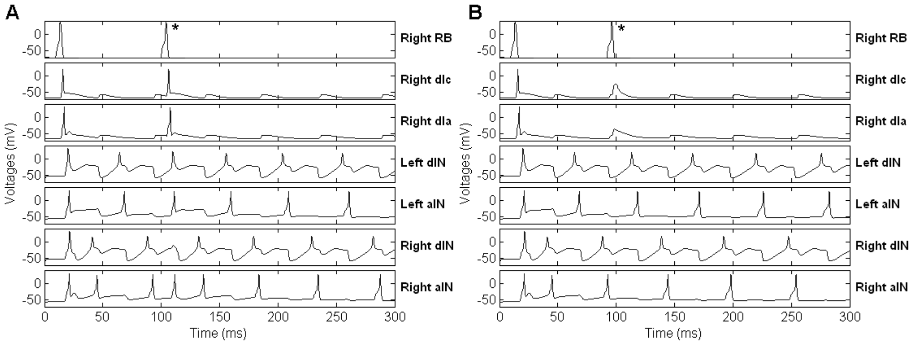


Figure 6.10: Effects of sensory gating during swimming. Two examples of swimming runs, in which the RB gets a second stimulus during swimming (*). **A**: The second RB spike causes the dlc and dla to fire, influencing the behaviour of the whole network (cINs and MNs omitted for clarity): the frequency of the following two cycles is increased. **B**: The RB spike now coincides with inhibition of the dlc and dla by the ipsilateral aIN. As a result, their firing is suppressed, sensory transmission is gated out and network behaviour is unaffected.

sulting from cell-cell recognition mechanisms, then the synapses that axons make, may be determined by the dendrites that they contact. This would mean that an axon grows, and simply connects to any neuron it encounters. The main reason for the differences in contact probability would then be caused by the dorsoventral position of cell bodies, dendrites and axons.

As will be shown in this section, the dendrites and axons of each neuron type have specific dorsoventral positions all along the tadpole spinal cord. Thus, developing axons will have a higher probability of connecting to those neurons, whose dendrites are in the same dorsoventral regions as the axons are growing. This could explain the differences in contact probability between neuron types.

For simplicity, in this section we will only consider neurons that are active during swimming activity.

First, by presenting results from paired recordings, we will show that there are indeed big differences in contact probability between different types of neuron. Secondly, anatomical evidence will be given on the distribution of axons and dendrites. These two experimental sections already strongly support the hypothesis of connections based on chance. Finally, in the modelling part, we explain how we construct probabilistic networks, based on the measurements from paired recordings and the resulting probabilities. Simulations will show that a network, constructed by creating connections purely relying on contact probabilities based on dorsoventral distributions, can produce good swimming output.

6.3.2 Anatomy of cells, axons and dendrites

As we have indicated before, the different types of neurons in the *Xenopus* spinal cord form loosely organized longitudinal columns. As in all vertebrates, the neurons form a dorsal to ventral sequence: sensory neurons; sensory interneurons; other interneurons; motoneurons. The neurons project axons into a marginal zone of longitudinal axons either directly or by first growing ventrally to the other side of the body and then turning or branching longitudinally.

RB neurons have a dorsal superficial soma and their central axons ascend and descend directly from the soma mainly in the dorsal area [68] to excite other neurons [91]. Dlcs have a multipolar soma (i.e. they have more than two processes emerging from the soma), in a dorsolateral superficial position. Their axon crosses the spinal cord ventrally to ascend in contralateral marginal zone (it may have a descending branch) to excite mainly motor circuit neurons [91]. Dlas, like dlcs, have a multipolar soma, in a dorsolateral superficial position. Their axons ascend directly from the soma to excite other mainly motor circuit neurons on the same side [92].

The other neurons (aIN, cIN, dIN and MN) all lie inside the marginal zone and are part of the motor circuit. AINs have a unipolar soma (i.e. they only have one process), with the dendrite emerging from the ventrally projecting initial segment of the axon, an axon that turns to ascend and then branches to give a descending axon. CINs have a unipolar soma, short radial dendrites from the thick initial segment of the axon, and a ventral commissural axon that ascends contralaterally or branches there to descend. DINs have a bipolar or multipolar soma, with dorsal and ventral dendrites, and an axon that descends [112] to excite other neurons [93]. MNs have a multipolar soma, with dendrites mainly ventral in the spinal

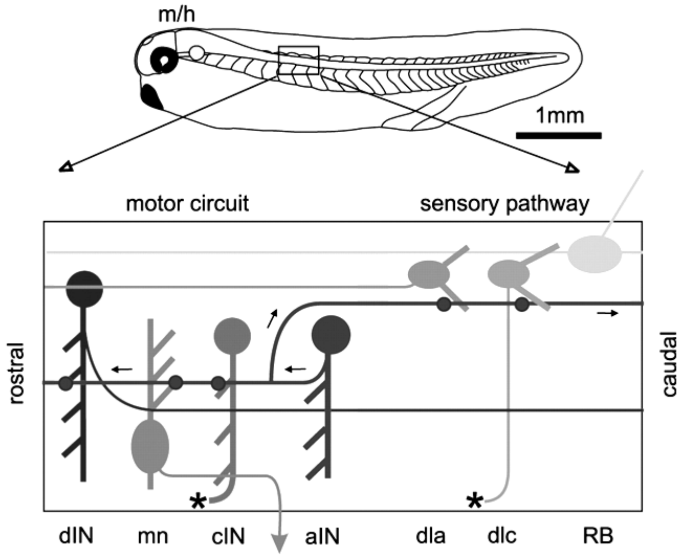


Figure 6.11: Schematic picture of the neurons and synapses in the *Xenopus* spinal cord. The diagram shows the different neuron types that are active during swimming, their position and the location of axons and dendrites.

cord, and an axon that projects caudally before exiting to innervate muscles [113]. Central axon synapses release acetylcholine to excite other motoneurons [106].

Figure 6.11, which was already included in Chapter 5 as Figure 5.4, shows in a schematic drawing the location of neurons, axons and dendrites in the *Xenopus* spinal cord.

6.3.3 Evidence from recordings on synaptic connections

At the *Xenopus* tadpole research group at the University of Bristol, current clamp recordings from over 500 pairs of neurons were made to investigate the synaptic connections between the seven different classes of spinal neuron in the swimming network [Li et al., unpublished]. By injecting current into each neuron to evoke an action potential it was possible to see if the neuron produced postsynaptic excitation or inhibition of the other. After recording, the animals were fixed and the neurons finally identified. Only those pairs with clear anatomical identification and where the axon of at least one neuron was seen to have a possible contact point onto the soma or dendrites of the other neuron were included in the analysis.

For some specific synaptic connections, paired recordings were very difficult to make either due to access difficulties (e.g. more ventral neurons like motoneurons are more difficult to expose), or because the presynaptic neurons are scarce (e.g. dla connections to motor circuit neurons), or because the probability of synaptic connection is low (e.g. dIN connections to dlc). In these cases, indirect methods were also used to assess connections in conjunction with any paired recordings available, e.g. use skin stimulation instead of current injection, block certain synaptic components to see if others are active, etc.

In this thesis, we will not go into detail on the different methods used, but Table 6.2 gives an overview of the results from the paired recordings. Overall, the results from paired recordings and other physiological recordings summarized in Table 6.2 show very widespread connectivity. Where evidence is available, neurons with dendrites receive synapses from all other neuron classes, although some connections are clearly more preferred than others. This was unexpected and raised the possibility that the formation of synaptic connections in the developing spinal cord may be more stochastic and not precisely determined by processes of cell to cell recognition.

6.3.4 Anatomical evidence on axonal and dendritic distribution

As we have explained, the probability of contact could depend mainly on the dorsoventral distribution of axons and dendrites. Therefore these distributions for the main spinal neuron classes in the rostral spinal cord were examined by the biologists in the *Xenopus* tadpole research group. By injecting marker dye in the neurons, the soma, dendrites and complete axonal projections could be seen and traced in whole mounts of the spinal cord viewed laterally (e.g. [89],[Li et al.,

Table 6.2: Synaptic connections found by paired recording. ’*’ indicates the cases where preliminary recording was used to select pairs of neurons that were connected, so probabilities of connection are not meaningful. For each neuron class, *ipsi* = synapses made ipsilaterally, and *contra* = synapses made contralaterally.

pre neuron	post neuron	RB	dlc	dla	aIN	cIN	dIN	MN	Total
RB	pairs	2	54	17	15	44	0	0	132
ipsi	synapses	0	34	6	2	4	0	0	46
	prob.	0	0.63	0.35	0.13	0.09			
dlc	pairs	0	1	0	6	42	3	13	65
contra	synapses	0	0	0	2	18	1	6	27
	prob.		0		0.33	0.43	0.33	0.46	
dla	pairs	17	10	0	8	12	2	0	49
ipsi	synapses	0	4?	0	2	1	0	0	7
	prob.	0	0.4		0.25	0.08	0		
aIN	pairs	15	21	8	16	39	10	3	112
ipsi	synapses	1	17	3	4	6	2	1	34
	prob.	0.07	0.81	0.38	0.25	0.15	0.2	0.33	
cIN	pairs	0	42	0	13	35	1	3	94
contra	synapses	0	0	0	0	9	1	0	10
	prob.		0	0	0	0.26	1	0	
dIN	pairs	0	0	2	7	7	62	32	110
ipsi	synapses	0	0	0	6	6	45	27	84
	prob.			0	*	*	*	*	

unpublished]).

The dorsoventral range of dendrites was determined from the positions of the most ventral and most dorsal dendrite for each neuron. The dendrite dorsoventral ranges were then summed for each neuron class (except for RB, which have no dendrites). For each 10% dorsoventral position bin, the probability was found that an individual neuron of each class would have dendrites in that bin (Table 6.3). The dendrite distributions for neurons active during swimming (MNs, aINs, cINs and dINs) were broad but all had a maximum just below the dorsoventral midline (in the 30% or 40% bin) and fell away dorsally. In contrast the dendrites of dlc sensory pathway interneurons had a maximum dorsally (in the 80% bin) and fell away ventrally.

The dorsoventral positions of axons were measured every 0.05 mm from the neuron soma and information was discarded about the distance of the axon from the soma and whether it was ascending or descending. For each neuron class we then summed these measures and calculated the probability of axons occupying different dorsoventral positions (Table 6.3). The dorsoventral axon distributions of some neurons are rather narrow. RB sensory neuron axons are dorsal (from 60% to 100%; maximum at 80%) while motoneurons are ventral (from 10% to 50%; maximum at 30%). dINs and dlcs are both slightly biased towards ventral positions (from 10% to 60% and 70% respectively; max at 30%-40%). The two inhibitory classes (aINs and cINs) have broader dorsoventral axon distributions (10% to 100%; max 30% and 40%-50% respectively).

Once the dorsoventral distributions of axons and dendrites were established, ’contact’ probabilities between axons and dendrites were calculated as follows for each pair of neuron classes. The probabilities of individual axons or dendrites occupying a particular dorsoventral region (10% dorsoventral position) were taken from their measured distributions, as the proportion found in each region. The probability that a particular presynaptic axon and postsynaptic dendrite would both occupy the same dorsoventral region was given by the product of their probabilities. Overall contact probabilities between each class of neuron were then found by adding up the separate probabilities for the 10 dorsoventral regions (Table 6.4). The contact probabilities range from 0.04 to 0.91 and relate intuitively

Table 6.3: Dorsoventral distribution of axons and dendrites of different neuron types. The dorsoventral range was subdivided in 10 bins, indicated by the percentages, where 0% is most ventral and 100% most dorsal. RBs have no dendrites, therefore that column is empty.

Axons	RB	dlc	aIN	cIN	dIN	MN
position	%	%	%	%	%	%
100	15	0	4	0	0	0
90	22	0	4	3	0	0
80	30	0	8	3	0	0
70	27	3	16	3	0	0
60	6	13	18	9	8	0
50	0	12	19	13	17	10
40	0	25	19	17	43	23
30	0	27	10	20	20	51
20	0	17	3	23	11	10
10	0	3	1	9	1	5
Dendrites		dlc	aIN	cIN	dIN	MN
position		%	%	%	%	%
100		3	0	0	0	0
90		28	5	1	3	0
80		31	7	6	5	0
70		21	9	11	16	2
60		14	11	17	18	6
50		3	14	18	18	13
40		0	16	23	21	19
30		0	14	17	11	21
20		0	14	6	5	21
10		0	11	0	3	17
	n = 14	n = 10	n = 9	n = 23	n = 10	n = 8

Table 6.4: Probability of axon of one neuron type ’contacting’ a dendrite of another neuron type.

	dendrites				
axons	dlc	aIN	cIN	dIN	MN
RB	0.65	0.39	0.23	0.32	0.04
dlc	0.082	0.86	0.61	0.60	0.81
aIN	0.30	0.73	0.57	0.63	0.50
cIN	0.12	0.82	0.50	0.50	0.77
dIN	0.12	0.91	0.70	0.69	0.84

to functions. They are higher for contacts from RB sensory neurons onto sensory pathway dlcs (0.65) than onto other neurons like dINs (0.32); they are low for dlc contacts with each other (0.08) but higher onto the neurons activated after skin stimulation (0.6 - 0.86 for aINs, cINs, dINs and MNs); they are quite high for contacts between neurons active during swimming (0.5 - 0.91 for aIN, cIN and dIN contacts to each other and to MNs).

6.3.5 Probabilistic network model

The results from recordings and anatomy together suggest that early spinal networks are able to develop using very simple rules. We therefore wanted to test if simple stochastic rules of connectivity could lead to functioning networks that can generate patterns of motor output suitable to produce swimming. Recent experiments have shown that a very small part of the spinal cord and hindbrain of the young tadpole, only 0.3 mm long, can generate long-lasting swimming after a brief stimulus [96]. Our aim was to model this minimal region of the nervous system.

We used the same individual neuron and synapse models as before, the ones described in Chapter 5. For synapses, we used similar conductances as in the swimming model that was explained in detail in Section 5.2. Now we used actual (though small) populations of neurons, instead of only single cells, and we wanted the contacts to be based on the data gathered in Table 6.4.

For simplicity we used a single sensory RB neuron which excited all sensory pathway dlc and dla INs on one side. Apart from one RB neuron, there were 10 instances of each of the six other types of neuron on both sides of the body, resulting in a network of 121 neurons. The network has left and right sides which inhibit each other to produce alternating swimming activity as described previously [114] (Figure 6.12).

The synaptic connections were probabilistic, in the sense that during construction, a random number (between 0 and 1) was generated, and a connection was actually made if this number was smaller than the probability from Table 6.4 for the types of neurons involved. To imitate more realistic numbers of neurons (30 of each type instead of 10), each model neuron had 3 chances to make a contact to another neuron, so synaptic strengths could be 0 or 1 to 3 times the single synapse strength. That way, each single neuron model actually represents 3 real neurons, which fire synchronously. When making these connections, the side of the body, and whether or not the axons of one type would cross the spinal cord, were taken into account (e.g. cINs were only allowed to connect to contralateral neurons, dINs only to ipsilateral, etc.). Another feature is that we made sure that when the probability of connection between two types is above 0, that there would always be at least 1 connection between two such neurons.

Because we were now using multiple instances of each neuron type, an extra type of connection had to be modelled, namely the electrical synapse. This is because ipsilateral MNs and dINs have electrical connections with each other with probability about 0.3 [106]. These electrical synapses are instantaneous, and modelled by adding the following term to the synapse model equations (5.8):

$$CS_{elec}(i,j):(V_i - V_j) \tag{6.1}$$

where V_i is the voltage of the presynaptic, and V_j of the postsynaptic neuron. The (dimensionless) connection strength CS_{elec} for these electrical synapses was set to 0.2. Since the capacitance was set to 4 for all neurons in the current network, this means that an increase in voltage in the presynaptic neuron of 10 mV, causes

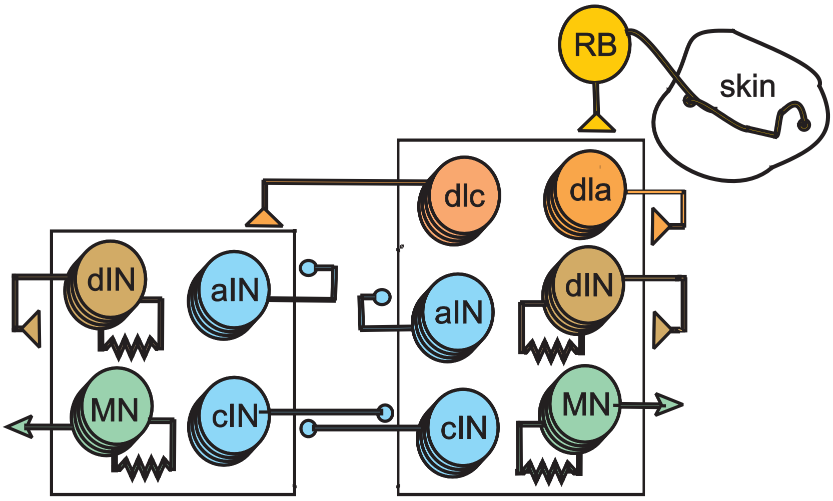


Figure 6.12: Schematic picture of the neurons and synapses in the model probability network. The diagram shows the different neuron types that are active during swimming, their chemical and electrical synapses. Triangles stand for excitatory synapses, circles for inhibitory synapses. A synapse to a 'box' means the connection is made to all neurons inside the box.

the same increase of 0.5 mV in the postsynaptic neuron. Note that these electrical synapses work in both directions, and can thus both increase and decrease the membrane potential in both neurons, depending on their relative charges.

A diagram of the resulting network is shown in Figure 6.12. After the construction of this network, simulations are run by giving an input pulse to the RB of the network. This resulted in reliable swimming behaviour in most cases, namely about 60% of the trials (n = 25). An example output of this swimming behaviour is shown in Figure 6.13. In most cases, when no swimming output was shown, this was due to a breakdown of spiking during the initiation process. In a single case, synchronous firing was seen on both sides of the body.

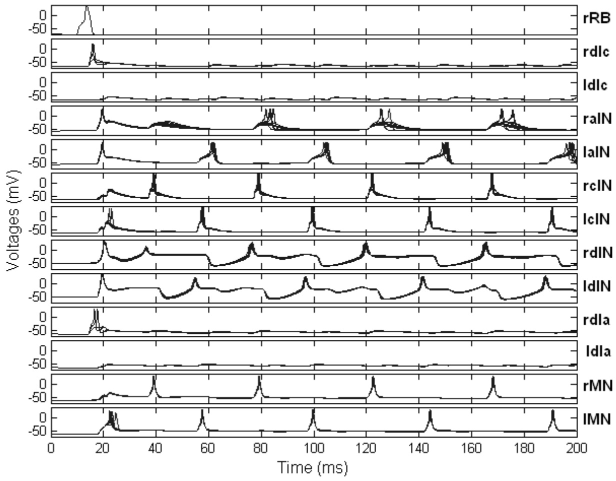


Figure 6.13: Example of reliable swimming output from the probabilistic network.

The synaptic strengths were reduced compared to the simple swimming network, to match the experimental maximum synaptic conductances. This was necessary, because each single synapse from the simple swimming network is replaced by several synapses in the probabilistic network. In this probabilistic network, the maximum synaptic conductances were computed by (at each timestep) adding up all incoming signals from one type of neurons to one specific neuron. This way, the resulting maximum value is comparable to the experimentally measured maximum conductances and to the conductances used in the simple swimming network. We optimized the synaptic strengths to match these conductances closely. Table 6.5 gives a comparison of the maximum conductances from the two networks, for the connections that are present in both. The biggest differences are in the dlc/dla-dIN excitation and the dIN-cIN excitation, which have higher maximum conductances in the probabilistic network. Early attempts to remove these differences by reducing synaptic strengths resulted in the failure to produce sustained swimming. In the case of the dlc/dla-dIN excitation, this may be due to the use of a single RB neuron, thus restricting the realism of the sensory pathway as a whole. In the case of dIN-cIN excitation, it is probably due to the cIN-model used, since in the simple swimming network, a stronger dIN-cIN excitation was also needed to get reliable swimming output.

The average of the contact probabilities in Table 6.4 is 0.6646. So this is the probability of any neuron contacting any other neuron in the network. If during network construction, we assign this probability to all connections and keep the overall degree of connectivity in the network constant, this would correspond to a purely random connection pattern of the neurons. We found that the resulting networks mostly failed in producing realistic swimmming behaviour. During a number of tests (n = 20), each time generating a new network, swimming was produced in only 20% of the trials.

Our simulations indicate that the network construction should not be considered as completely random, but that the constraints on construction could be based purely on location and distribution of neurons, axons and dendrites. This appears to be sufficient to create a network that produces swimming behaviour fairly reliably.

Table 6.5: Maximum synaptic conductances during swimming for the simple swimming model and the probability model.

		simple model		**probability model**	
source	**target**	**AMPA**	**NMDA**	**AMPA**	**NMDA**
RB	**dlc**	6.77	0.16	7.288	0.159
	dla	3.38	0.16	3.887	0.159
dlc	**dIN**	2.92	0.96	8.885	2.261
dla	**dIN**	2.55	1.08	4.198	1.777
dIN	**aIN**	1.46	0.54	1.915	0.609
	cIN	8.51	1.05	12.567	1.294
	dIN	4.38	6.86	4.261	6.389
	MN	3.89	1.47	4.356	1.570
		simple model		**probability model**	
		Inhibition		**Inhibition**	
cIN	**aIN**	2.86		2.92	
	cIN	2.86		2.92	
	dIN	9.53		9.466	
	MN	3.81		3.94	
aIN	**cIN**	1.11		3.092	
	dIN	2.70		1.93	
	MN	1.27		4.229	
	dlc	6.36		5.427	
	dla	6.36		4.229	

Chapter 7

Future work

This chapter elaborates on plans for MATCONT, for the use of MATCONT for cell-cycle modelling, and for creating a MATCONT tutorial. On the level of tadpole modelling, there are already plans for a network to model the struggling behaviour, and then combine the networks into one. We also hope to use NEST, in C^{++}-code, for simulations, to be able to increase the number of neurons and add length dimension to the model. Another idea is to look at reduced models to simulate cells.

7.1 Plans on MATCONT

7.1.1 The homotopy method

For the initiation of homoclinic orbits, there is one method that has not yet been implemented in MATCONT, but is currently under development. This so-called homotopy method constructs a homoclinic-to-saddle orbit starting from only the unstable equilibrium.

As described in Section 3.3.2, the defining equations for a homoclinic orbit can be written as

$$\dot{x}(t) - 2Tf(x(t), \alpha) = 0 \tag{7.1}$$

$$f(x_0, \alpha) = 0 \tag{7.2}$$

$$\int_0^1 \dot{\widetilde{x}}^*(t)[x(t) - \widetilde{x}(t)]dt = 0 \tag{7.3}$$

$$Q_{U^\perp}(s)^T(x(0) - x_0) = 0 \tag{7.4}$$

$$Q_{S^\perp}(s)^T(x(1) - x_0) = 0 \tag{7.5}$$

$$\|x(0) - x_0\| - \epsilon_0 = 0 \tag{7.6}$$

$$\|x(1) - x_0\| - \epsilon_1 = 0 \; . \tag{7.7}$$

We replace equations (7.6) and (7.7) by equivalent ones:

$$x(0) = x_0 + \epsilon_0 d_0 \tag{7.8}$$

$$x(1) = x_0 + \epsilon_1 d_1 \; , \tag{7.9}$$

where $\|d_0\| = \|d_1\| = 1$.

And we remove the constraint that the last vector is in the stable eigenspace of the equilibrium, by replacing (7.5) by:

$$\tau_j = \langle d_1, q_{1,j} \rangle \; , j = 1, ..., n_1 \; , \tag{7.10}$$

where the $q_{1,j}$ are the vectors of $Q_{S^\perp}(s)^T$, and we are keeping the τ_j variable, instead of demanding them to be zero.

The construction process of a homoclinic orbit fitting the above equations, from an unstable equilibrium, can be split up into several confined steps.

Initially, we replace (7.8) by

$$x(0) = x_0 + \epsilon_0 \sum_{j=1}^{n_0} c_j q_{0j} \tag{7.11}$$

$$\sum_{j=1}^{n_0} c_j^2 = 1 \; . \tag{7.12}$$

As initialization of the problem we set T and ϵ_0 small, and d_0 the leading unstable eigenvector of the equilibrium,

$$x(t) = x_0 + \epsilon_0 d_0, 0 \leq t \leq 1 \; , \tag{7.13}$$

$\epsilon_1 = |x(1) - x_0|$, $d_1 = (x(1) - x_0)/\epsilon_1$, $c_1 = 1$ and $c_2 = ... = c_{n_0} = 0$.

This initial curve is then extended by doing a continuation experiment with respect to the period T and ϵ_1, while keeping all c_j fixed. During this first continuation, ϵ_0 and d_0 are kept fixed. ϵ_1 should first be increasing, but then start to decrease again. Once this decrease stops, the whole continuation is stopped, and the orbit segment corresponding to the found solution at that point is kept as initial approximation of the homoclinic orbit.

After this initialization, the orbit will be found by locating zero intercepts of the τ_j in (7.10), one at a time. E.g. first keep T fixed and free c_1 and c_2. Thus, we let the initial vector between x_0 and $x(0)$ vary within a wider subspace of the unstable eigenspace. This way we try to find a zero of τ_1. In following steps, we always fix T and all τ_j that are zero already, and try to locate a zero of the next τ_i, while each time freeing more c_j to replace the fixed τ_j.

Finally, the orbit segment obtained is considered a good approximation of the homoclinic orbit. So then system parameter λ is set free, and by the Newton correction method used at the start-up of any continuation, the approximated orbit segment should converge to the exact homoclinic orbit.

We plan to include this whole process in MATCONT (we are currently working on it), which will involve all consecutive steps, changing Graphical User Interface (GUI) windows, introducing different variables and test functions, etc.

7.1.2 User-related issues

To accommodate to user remarks, we plan to rewrite the MATCONT tutorial and help file, and include all updates, new features, etc. Now many users have to be helped by e-mail, one by one, which is a time-consuming occupation.

There are also several other features that could be added, as suggested by users:

- Support for general MATLAB ODE m-files as right-hand sides.
- Support for Simulink and SimTech models, which are two MATLAB toolboxes which are often used in engineering.

- Allow more of the standard MATLAB graphics handling in MATCONT's plotting windows.
- Improve the organisation of the GUI in general, to allow further extensions more easily.

7.1.3 Modelling of the cell cycle

In January a new project will start in our department, in collaboration with people of the VIB, a biotechnological research group at Ghent University. The aim of the project is to carry out detailed studies of the cell cycle regulatory processes, more precisely the cell cycle of the model plant *Arabidopsis*. The cell cycle, during which the cell divides into several identical cells, is an important biological process that is governed by the dynamic regulating influence of many genes.

This project will involve a combination of different simulation approaches, that include the use of ordinary differential equations (ODEs). For this MATCONT and CL_MATCONT will be used. Using these packages, detailed analyses of key switching mechanisms of the cell cycle will be performed, with a focus on events associated with state transitions (e.g. from resting to actively dividing), i.e. bifurcations.

For some processes of which less is known, piecewise linear differential equations (PLDEs) may be used for simulation. And a possible hybrid system, combining both simulation methods, would allow to fully use the available data.

But the core of the project will be the use of ODEs. New algorithms will have to be developed to model key cell cycle checkpoints, which will help to study certain non-linear phenomena in the process. These algorithms will be implemented in MATCONT. Later, these models will be generalized, to be efficient for multiple system predictions and simulations, and refined for a clear parameter structure for specific applications. The resulting ODE modelling platform should then be integrated in a framework to be used by computational biologists, for future modelling in their field of research.

The ultimate aim is the integration of various modelling strategies that can be used for the simulation of biological networks. The project will hopefully result in a powerful modelling platform to gain understanding of the mechanisms and processes that govern the (plant) cell cycle.

7.2 Plans on *Xenopus* tadpole research

7.2.1 Struggling network

Most animals have a range of different motor patterns, like walking, trotting, running and scratching. In mammals the spinal neuronal circuits used to select and produce different locomotor gaits and to switch from one to the other are unclear. It has often been suggested that neuromodulation is involved in motor pattern selection.

Since all vertebrates have a common ancestry, the basic mechanisms of motor pattern selection have been studied in lower vertebrates like young *Xenopus* tadpoles. Hatchling tadpoles respond in two main ways. When touched, they swim off using low amplitude, alternating waves of bending at 15 to 25 Hz that travel from the head towards the tail [11]. When held, they struggle producing large amplitude, alternating bends at 5 to 10 Hz that travel from the tail towards the head [123]. Our aim is to uncover the cellular neuronal mechanisms that allow different patterns of sensory activation to select the different motor patterns of swimming and struggling.

To find out how struggling is generated, the whole-cell patch method [94] was used to record from hindbrain and spinal neurons during 40 Hz stimulation of the skin. Electrical stimulation to the trunk skin on one side of the body at 40Hz imitates applied pressure and can lead to struggling activity in the motoneurons on both sides [123]. The recordings suggested that there are some types of neuron that become active during 40 Hz stimulation and that are not active during swimming, both in the sensory pathway and in the CPG [Li et al., unpublished]. We now need to ask whether these new neurons can connect in such a way with the other neurons, which we have already modelled in detail, to form a network capable of generating struggling activity during repeated sensory stimulation to the skin on one side.

Early versions of models for these neurons have already been developed and tested in a small struggling network. At first, tests were done to see if aINs could be responsible for the burst termination during struggling, but it appeared not to be reliable enough for robust alternation of the bursts. Further tests have indicated that a new mechanism, i.e. synaptic depression, is needed for the alternation of bursts to happen reliably. This synaptic depression means that when spikes are fired at high frequency in the presynaptic neuron, the synapse will gradually lose conductance, and the postsynaptic effect will weaken. Experiments have supported this idea of synaptic depression, by showing that reciprocal connections between cINs and contralateral CPGs show no significant depression at 20 Hz, a typical swimming frequency, but they became significantly depressed at 100 Hz, a typical frequency at which cINs fire during struggling bursts. The cINs, which prevent the contralateral neurons to fire during a burst, will thus become less effective in suppressing as the burst goes on, and eventually be unable to prevent the opposite side to become active, which will in his turn suppress spiking in the first side, etc.

We want to improve the individual neuron models of these two types of struggling-specific neurons, incorporate them in the struggling network, to get an overall good struggling network, including the sensory pathway, as we did for swimming.

A further goal is then to combine the swimming and struggling networks into one whole network, which will respond with swimming output to a single stimulus, and with bursting output to high-frequency stimulation.

7.2.2 Population models using NEST

The network models that we have developed, even the probabilistic model, have a relatively small number of neurons. The basic swimming model has only 11 neurons, the probabilistic model has 121. To build a neural network with a realistic number of neurons (1000-2000) we will need to step away from the present framework. MATLAB is too slow as programming language, and the framework is too restricted to handle large numbers of neurons.

A solution may lie in the NEST initiative, NEural Simulation Technology (`http://www.nest-initiative.org/`). NEST is a simulation system for large networks of biologically realistic neurons. The NEST initiative is a collaboration, founded in 2001, between researchers at the Honda Research Institute Europe in Freiburg (Germany), the Albert-Ludwigs Universität Freiburg (Germany), the Norwegian University of Life Sciences in As (Norway) and the Max-Planck Institut für Strömungsforschung in Göttingen (Germany).

NEST is, according to the developers, ’best suited for the simulation of large networks of spiking point-neuron models, of which the internal dynamics may be arbitrarily complex’. This is indeed what we have been working with: single-compartment models which we want to combine in large networks. And the main focus of NEST simulations is ’the dynamics and behaviour of large networks of neurons’, which is exactly what we want to investigate. NEST is written in C^{++} and supposedly completely platform-independent.

A new version of NEST, which would become available early 2007, includes all functionalities that we have needed so far: different types of synapses with many features, including delays and synaptic depression. Output is stored in a very general format, which can be processed at will by the user.

Thus, it seems like an ideal tool to use in the future steps of our network modelling research.

7.2.3 Addition of length dimension

Once we are able to model neural networks with realistic numbers of neurons, we can step away from the modelling of a single segment of the spinal cord. Using software like NEST, we hope to be able to add a length dimension to our network model, with the necessary delays in the synaptic connections, depending on the axon lengths. When this can be incorporated into the model, it becomes possible to see the swim cycle move down, and the struggle cycle move up the body, and see the flexing of the *Xenopus* tadpole body.

But to do that, we will also need to have a better grasp on the effect of axon length on the synaptic signal. So far, we have always used standard synapses, with fixed delays, without adding a relation to axon length and signal strength or behaviour. This will require some further research and modelling of axonal effects.

7.2.4 Theoretical study using reduced models

A very different idea is to step back down from the biologically realistic neuron models, with ionic currents and conductances, to reduced models. These are neuron models, defined by dynamical systems, but for which the parameters have no direct biological meaning. The advantage of these models is that they are much more concise and easier to integrate numerically. It would be interesting to investigate whether the network behaviour (both in simple networks, swimming and struggling, as in a possible large network with thousands of neurons) can be modelled to a comparable accuracy when using such reduced neuron models instead of the biologically realistic ones.

Index

Bibliography

[1] S.P. Aiken, F.M. Kuenzi and N. Dale, Xenopus embryonic spinal neurons recorded in situ with patch-clamp electrodes–conditional oscillators after all? Eur. J. Neurosci. 18 (2003), 333-343.

[2] V.I. Arnol'd, Geometrical methods in the theory of ordinary differential equations. Springer-Verlag, New York (1983).

[3] U. Ascher and R.D. Russell, Reformulation of boundary value problems in standard form. SIAM Rev. 23(2) (1981), 238-254.

[4] A.D. Bazykin, Mathematical biophysics of interacting populations. (In Russian) Nauka, Moscow (1985).

[5] A.D. Bazykin, Nonlinear dynamics of interacting populations. World Scientific, Singapore (1998).

[6] H. Beresticky and L. Nirenberg, Traveling fronts in cylinders. Ann. Inst. Henri Poincaré 9(5) (1992), 497-572.

[7] A. Beuter, L. Glass, M.C. Mackey and M.S. Titcombe, Nonlinear Dynamics in Physiology and Medicine. Springer Interdisciplinary Applied Mathematics Vol. 25 (2003).

[8] W.-J. Beyn, A. Champneys, E. Doedel, W. Govaerts, Yu.A. Kuznetsov and B. Sandstede, Numerical continuation, and computation of normal forms. In

B. Fiedler (ed.), Handbook of Dynamical Systems, Vol. 2, Ch. 4. Elsevier Science (2002).

[9] D. Bindel, J. Demmel and M. Friedman, Continuation of invariant subspaces for large bifurcation problems. In proceedings, SIAM Conference on Applied Linear Algebra, July 2003, `http://www.siam.org/meetings/la03/proceedings/`.

[10] I.I. Blechman, Synchronization of dynamical systems [in Russian: ”Sinchronizatzia dinamicheskich sistem”]. Nauka, Moscow (1971).

[11] K.M. Boothby and A. Roberts, The stopping response of Xenopus laevis embryos, behaviour, development and physiology. J. Comp. Physiol. A 170 (1992), 171-180.

[12] H. Broer and G. Vegter, Subordinate Shil'nikov bifurcations near some singularities of vector fields having low codimension. Ergod. Theor. Dyn. Syst. 4 (1984), 509-525.

[13] E. Brown, J. Moehlis and P. Holmes, On the phase reduction and response dynamics of neural oscillator populations. Neural Comput. 16 (2004), 673-715.

[14] A.R. Champneys and Yu.A. Kuznetsov, Numerical detection and continuation of codimension-two homoclinic bifurcations. Int. J. Bifurcation Chaos 4 (1994), 785-822.

[15] A.R. Champneys, Yu.A. Kuznetsov and B. Sandstede, A numerical toolbox for homoclinic bifurcation analysis. Int. J. Bifurcation Chaos 6(5) (1996), 867-887.

[16] D.E. Chapman, K.A. Keefe and K.S. Wilcox, Evidence for functionally distinct synaptic NMDA receptors in ventromedial versus dorsolateral striatum. J. Neurophysiol. 89 (2003), 69-80.

[17] T.R. Clandinin and S.L. Zipursky, Making connections in the fly visual system. Neuron 35 (2002), 827-841.

[18] J.D.W. Clarke, B.P. Hayes, S.P. Hunt and A. Roberts, Sensory physiology, anatomy and immunohistochemistry of Rohon-Beard neurones in embryos of Xenopus laevis. J. Physiol. 348 (1984), 511-525.

[19] K. Cole, Membranes, ions and impulses. University of California Press, Berkeley (1968).

[20] J.A. Connor, D. Walter and R. McKown, Modifications of the Hodgkin-Huxley axon suggested by experimental results from crustacean axons. Biophys. J. 18 (1977), 81-102.

[21] S.M. Crook, G.B. Ermentrout and J.M. Bower, Spike frequency adaptation affects the synchronization properties of networks of cortical oscillators. Neural Comput. 10 (1998), 837-854.

[22] N. Dale, Reciprocal inhibitory interneurones in the Xenopus embryo spinal cord. J. Physiol. 363 (1985), 61-70.

[23] N. Dale, Alan Roberts, O.P. Ottersen and J. Storm-Mathisen, The morphology and distribution of 'Kolmer-Agduhr cells', a class of cerebrospinal-fluid-contacting neurons revealed in the frog embryo spinal cord by GABA immunocytochemistry. Proc. R. Soc. Lond. B Biol. Sci. 232 (1987), 193-203.

[24] N. Dale, Kinetic characterization of the voltage-gated currents possessed by *Xenopus* embryo spinal neurons. J. Physiol. 489(2) (1995), 473-488.

[25] N. Dale, Experimentally derived model for the locomotor pattern generator in the *Xenopus* embryo. J. Physiol. 489(2) (1995), 489-510.

[26] N. Dale, Coordinated motor activity in simulated spinal networks emerges from simple biologically plausible rules of connectivity. J. Comput. Neurosci. 14 (2003), 55-70.

[27] C. De Boor and B. Swartz, Collocation at Gaussian points. SIAM J. Numer. Anal. 10 (1973), 582-606.

[28] A. Demir, D. Long and J. Roychowdhury, Computing Phase Noise Eigenfunctions Directly From Steady-State Jacobian Matrices. ACM/IEEE Int. Conf. on Computer-Aided Design (2000).

[29] J.W. Demmel, L. Dieci, and M.J. Friedman, Computing connecting orbits via an improved algorithm for continuing invariant subspaces. SIAM J. Sci. Comput. 22(1) (2001), 81-94.

[30] P. Deuflhard and A. Hohmann, Numerische Mathematik: Eine algorithmisch orientierte Einführung. Walter de Gruyter, New York (1991).

[31] A. Dhooge, W. Govaerts, Yu.A. Kuznetsov, W. Mestrom and A.M. Riet, A continuation toolbox in Matlab. Manual, `http://www.matcont.ugent.be/doc_cl_matcont.zip` (2003).

[32] A. Dhooge, W. Govaerts and Yu.A. Kuznetsov, MatCont: A Matlab package for numerical bifurcation analysis of ODEs. ACM TOMS 29(2) (2003), 141-164.

[33] A. Dhooge, B. Sautois, W. Govaerts and Yu.A. Kuznetsov, MatCont: Matlab software for bifurcation study of dynamical systems. ICNAAM 2005 proceedings (2005), 838-841.

[34] A. Dhooge, MatCont: MATLAB software for bifurcations of dynamical systems. PhD Thesis, Ghent University (2005).

[35] A. Dhooge, W. Govaerts, B. Sautois, Yu.A. Kuznetsov and H. Meijer, New features of the software MatCont for bifurcation analysis of dynamical systems. Submitted to ACM TOMS (2006).

[36] L. Dieci and T. Eirola, On smooth decomposition of matrices. SIAM J. Matrix Anal. Appl. 20 (1999), 800-819.

[37] O. Diekmann and J.A.P. Heesterbeek, Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation. Wiley Series in Mathematical and Computational Biology (2000).

[38] E.J. Doedel, A.R. Champneys, T.F. Fairgrieve, Yu.A. Kuznetsov, B. Sandstede and X.J. Wang, Auto97: Continuation and bifurcation software for ordinary differential equations (with HomCont). `http://indy.cs.concordia.ca/auto`, Concordia University, Montreal, Canada (1997-2000).

[39] E.J. Doedel and M.J. Friedman, Numerical computation of heteroclinic orbits. J. Comp. Appl. Math. 26 (1989), 155-170.

[40] F. Dumortier, R. Roussarie, J. Sotomayor and H. Zoladek, Bifurcations of planar vector fields: Nilpotent singularities and Abelian integrals. Springer-Verlag, Berlin (1991).

[41] J. Emsley, The Elements of Murder. A History of Poison. Oxford University Press, Oxford, UK (2005).

[42] G.B. Ermentrout, $n : m$ phase-locking of weakly coupled oscillators. J. Math. Biol. 12 (1981), 327-342.

[43] G.B. Ermentrout and N. Kopell, Multiple pulse interactions and averaging in systems of coupled neural oscillators. J. Math. Biol. 29 (1991), 195-217.

[44] G. Ermentrout, Type I membranes, phase resetting curves and synchrony. Neural Comput. 8(5) (1996), 979-1001.

[45] G.B. Ermentrout, M. Pascal and B. Gutkin, The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators. Neural Comput. 13(6) (2001), 1285-1310.

[46] G. Ermentrout, Simulating, analyzing, and animating dynamical systems: A guide to XXPAUT for researchers and students. SIAM, Philadelphia (2002).

[47] C.P. Fall, E.S. Marland, J.M. Wagner and J.J. Tyson, Computational Cell Biology. Springer-Verlag, New York (2002).

[48] M. Fischberg, J.B. Gurdon and T.R. Elsdale, Nuclear transplantation in Xenopus laevis. Nature 181 (1958), 424.

[49] M. Friedman, W. Govaerts, Yu.A. Kuznetsov and B. Sautois, Continuation of homoclinic orbits in Matlab. LNCS 3514 (2005), 263-270.

[50] M. Galarreta and S. Hestrin, Properties of GABA(A) receptors underlying inhibitory synaptic currents in neocortical pyramidal neurons. J. Neurosci. 17 (1997), 7220-7227.

[51] W. Gerstner and W. Kistler, Spiking neuron models. Cambridge University Press, Cambridge (2002).

[52] L. Glass and M.C. Mackey, From clocks to chaos: the rhythms of life. Princeton University Press, Princeton (1988).

[53] L. Glass, Resetting and entraining biological rhythms. Ch. 5 in A. Beuter, L. Glass, M.C. Mackey and M.S. Titcombe, Nonlinear dynamics in physiology and medicine. Springer-Verlag, New York (2003).

[54] G.H. Golub and C.F. Van Loan, Matrix Computations. The Johns Hopkins University Press, Baltimore (1996).

[55] T. Gotz, U. Kraushaar, J. Geiger, J. Lubke, T. Berger and P. Jonas, Functional properties of AMPA and NMDA receptors expressed in identified types of basal ganglia neurons. J. Neurosci. 17 (1997), 204-215.

[56] W. Govaerts, Numerical Methods for Bifurcations of Dynamical Equilibria. SIAM, Philadelphia (2000).

[57] W. Govaerts and B. Sautois, Bifurcation software in Matlab with applications in neuronal modeling. Comput. Meth. Prog. Bio. 77(2) (2005), 141-153.

[58] W. Govaerts and B. Sautois, The onset and extinction of neural spiking: a numerical bifurcation approach. J. Comput. Neurosci. 18(3) (2005), 273-282.

[59] W. Govaerts and B. Sautois, Computation of the phase response curve: a direct numerical approach. Neural Comput. 18(4) (2006), 817-847.

[60] W. Govaerts and B. Sautois, Phase response curves, delays and synchronization in Matlab. LNCS 3992 (2006), 391-398.

[61] P. Gray and S.K. Scott, Chemical oscillations and instabilities: non-linear chemical kinetics. Oxford University Press, Oxford, UK (1990).

[62] C.S. Green and S.R. Soffe, Transitions between two different motor patterns in *Xenopus* embryos. J. Comp. Physiol. 178 (1996), 279- 291.

[63] S. Grillner and P. Wallen, Cellular bases of a vertebrate locomotor system-steering, intersegmental and segmental co-ordination and sensory control. Brain Res. Rev. 40 (2002), 92-106.

[64] J. Guckenheimer, Isochrons and phaseless sets. J. Math. Biol. 1 (1975), 259-273.

[65] M.R. Guevara, L. Glass, M.C. Mackey and A. Shier, Chaos in neurobiology. IEEE Trans. Syst. Man Cybern. 13(5) (1983), 790-798.

[66] D. Hansel, G. Mato and C. Meunier, Phase dynamics for weakly coupled Hodgkin-Huxley neurons. Neurophys. Lett. 23(5) (1993), 367-372.

[67] D. Hansel, G. Mato and C. Meunier, Synchrony in excitatory neural networks. Neural Comput. 7 (1995), 307-337.

[68] V. Hartenstein, Early pattern of neuronal differentiation in the Xenopus embryonic brainstem and spinal cord. J. Comp. Neurol. 328 (1993), 213-231.

[69] J. Hellgren, S. Grillner and A. Lansner, Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons. Biol. Cybern. 68(1) (1992), 1-13.

[70] S.I. Higashijima, G. Mandel and J.R. Fetcho, Distribution of prospective glutamatergic, glycinergic, and GABAergic neurons in embryonic and larval zebrafish. J. Comp. Neurol. 480 (2004), 1-18.

[71] S.I. Higashijima, M. Schaefer and J.R. Fetcho, Neurotransmitter properties of spinal interneurons in embryonic and larval zebrafish. J. Comp. Neurol. 480 (2004), 19-37.

[72] A.L. Hodgkin, The local electric changes associated with repetitive action in a non-medullated axon. J. Physiol. 107 (1948), 165-181.

[73] A.L. Hodgkin and A.F. Huxley, Action potentials recorded from inside a nerve fibre. J. Physiol. 144(1939), 710-711.

[74] A.L. Hodgkin and A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117 (1952), 500-544.

[75] F.C. Hoppensteadt and E.M. Izhikevich, Weakly connected neural networks. Springer-Verlag, New York (1997).

[76] C. Hos, A. Champneys and L. Kullmann, Bifurcation analysis of surge and rotating stall in the Moore-Greitzer compression system. IMA J. App. Math. 68 (2003), 205-228.

[77] E.M. Izhikevich, Neural excitability, spiking and bursting. Int. J. Bifurcation Chaos 10(6) (2000), 1171-1266.

[78] E.M. Izhikevich, Dynamical systems in neuroscience: the geometry of excitability and bursting. To be published by the MIT press (2006).

[79] D. Jamieson and Alan Roberts, Responses of young Xenopus laevis tadpoles to light dimming, possible roles for the pineal eye. J. Exp. Biol. 203 (2000), 1857-1867.

[80] J.A. Kahn and Alan Roberts, The neuromuscular basis of rhythmic struggling movements in embryos of Xenopus laevis. J. Exp. Biol. 99 (1982), 197-205.

[81] J.A. Kahn, Alan Roberts and S.M. Kashin, The neuromuscular basis of swimming movements in embryos of the amphibian Xenopus laevis. J. Exp. Biol. 99 (1982), 175-184.

[82] A. Khibnik, B. Krauskopf and C. Rousseau, Global study of a family of cubic Liénard equations. Nonlinearity 11 (1998), 1505-1519.

[83] M. Koper, Bifurcations of mixed-mode oscillations in a three-variable autonomous Van der Pol-Duffing model with a cross-shaped phase diagram. Phys. D 80 (1995), 72-94.

[84] J.H. Kotaleski, S. Grillner and A. Lansner, Neural mechanisms potentially contributing to the intersegmental phase lag in lamprey. Biol. Cybern. 81(4) (1999), 317-330.

[85] Yu.A. Kuznetsov and V.V. Levitin, Content: Integrated environment for analysis of dynamical systems, version 1.5. `http://ftp.cwi.nl/CONTENT`, CWI, Amsterdam (1998).

[86] Yu. A. Kuznetsov, Elements of Applied Bifurcation Theory (3rd ed.). Appl. Math. Sci. 112, Springer-Verlag, New York (2004).

[87] Yu. A. Kuznetsov, W. Govaerts, E.J. Doedel and A. Dhooge, Numerical periodic normalization for codim 1 bifurcations of limit cycles. SIAM J. Numer. Anal. 43(4) (2005), 1407-1435.

[88] Yu. A. Kuznetsov, Practical computation of normal forms on center manifolds at degenerate Bogdanov-Takens bifurcations. Int. J. Bifurcation Chaos 15(11) (2005), 3535-3546.

[89] W.-C. Li, R. Perrins, S.R. Soffe, M. Yoshida, A. Walford, Alan Roberts, Defining classes of spinal interneuron and their axonal projections in hatchling Xenopus laevis tadpoles. J. Comp. Neurol. 441 (2001), 248-265.

[90] W.-C. Li, S.R. Soffe and Alan Roberts, Spinal inhibitory neurons that modulate cutaneous sensory pathways during locomotion in a simple vertebrate. J. Neurosci. 22 (2002), 10924-10934.

[91] W.-C. Li, S.R. Soffe and Alan Roberts, The spinal interneurons and properties of glutamatergic synapses in a primitive vertebrate cutaneous flexion reflex. J. Neurosci. 23(27) (2003), 9068-9077.

[92] W.-C. Li, S.R. Soffe and Alan Roberts, Dorsal spinal interneurons forming a primitive, cutaneous sensory pathway. J. Neurophysiol. 92(2) (2004), 895-904.

[93] W.-C. Li, S.R. Soffe and Alan Roberts, Glutamate and acetylcholine corelease at developing synapses. PNAS 101 (2004), 15488-15493.

[94] W.-C. Li, S.R. Soffe and Alan Roberts, A direct comparison of whole cell patch and sharp electrodes by simultaneous recording from single spinal neurons in frog tadpoles. J. Neurophysiol. 92(1) (2004), 380-386.

[95] W.-C. Li, S. Higashijima, D.M. Parry, Alan Roberts and S.R. Soffe, Primitive roles for inhibitory interneurons in developing frog spinal cord. J. Neurosci. 24 (2004), 5840-5848.

[96] W.-C. Li, S.R. Soffe, E. Wolf and Alan Roberts, Persistent responses to brief stimuli: feedback excitation among brainstem neurons. J. Neurosci. 26 (2006), 4026-4035.

[97] J.E. Lisman, J.M. Fellous and X.J. Wang, A role for NMDA-receptor channels in working memory. Nat. Neurosci. 1 (1998), 273-275.

[98] I.G. Malkin, Methods of Poincaré and Lyapunov in the theory of non-linear oscillations [in Russian:"Metodi Puankare i Liapunova v teorii nelineinix kolebanii"]. Gostexizdat, Moscow (1949).

[99] I.G. Malkin, Some problems in nonlinear oscillation theory [in Russian: "Nekotorye zadachi teorii nelineinix kolebanii"]. Gostexizdat, Moscow (1956).

[100] D. McCormick, Membrane potential and action potential. In: M.J. Zigmond, F.E. Bloom, S.C. Landis, J.L. Roberts and L.R. Squire (eds.), Fundamental Neuroscience, Academic Press, New York (1999).

[101] C. Morris and H. Lecar, Voltage oscillations in the barnacle giant muscle fiber. Biophys. J. 35 (1981), 193-213.

[102] P.D. Nieuwkoop and J. Faber, Normal tables of Xenopus laevis (Daudin). North Holland, Amsterdam (1956).

[103] L. Nowak, P. Bregestovski, P. Ascher, A. Herbet and A. Prochiantz, Magnesium gates glutamate-activated channels in mouse central neurones. Nature 307 (1984), 462-465.

[104] M.A. Nowak and R.M. May, Virus dynamics: Mathematical Principles of Immunology and Virology. Oxford University Press, Oxford, UK (2000).

[105] D. Parker, Spinal-cord plasticity: independent and interactive effects of neuromodulator and activity-dependent plasticity. Mol. Neurobiol. 22 (2001), 55-80.

[106] R. Perrins and Alan Roberts, Cholinergic and electrical synapses between synergistic spinal motoneurones in the Xenopus laevis embryo. J. Physiol. (London) 485(1) (1995), 135-144.

[107] R. Perrins and S.R. Soffe, Local effects of glycinergic inhibition in the spinal cord motor systems for swimming in amphibian embryos. J. Neurophysiol. 76 (1996), 1025-1035.

[108] A.B. Ribera and N.C. Spitzer, Differentiation of IKa in amphibian spinal neurons. J. Neurosci. 10(6) (1990), 1886-1891.

[109] A.B. Ribera and D.A. Nguyen, Primary sensory neurons express a Shaker-like potassium channel gene. J. Neurosci. 13 (1993), 4988-4996.

[110] J. Rinzel and B. Ermentrout, Analysis of Neural Excitability and Oscillations. In: C. Koch and I. Segev (eds.), Methods in neuronal modeling: from synapses to networks, MIT Press, Cambridge (1989), 135-169.

[111] Alan Roberts, N. Dale, W.H. Evoy and S.R. Soffe, Synaptic potentials in motoneurons during fictive swimming in spinal Xenopus embryos. J. Neurophysiol. 54 (1985), 1-10.

[112] Alan Roberts and S.T. Alford, Descending projections and excitation during fictive swimming in *Xenopus* embryos: neuroanatomy and lesion experiments. J. Comp. Neurol. 250 (1986), 253-261.

[113] Alan Roberts, A. Walford, S.R. Soffe and M. Yoshida, Motoneurons of the axial swimming muscles in hatchling *Xenopus* tadpoles: features, distribution, and central synapses. J. Comp. Neurol. 411 (1999), 472-486.

[114] Alan Roberts and M.J. Tunstall, Mutual re-excitation with post-inhibitory rebound: a simulation study on the mechanisms for locomotor rhythm generation in the spinal cord of the *Xenopus* embryos. Eur. J. Neurosci. 2 (1990), 11-23.

[115] Alan Roberts, Early functional organization of spinal neurons in developing lower vertebrates. Brain Res. Bull. 53 (2000), 585-593.

[116] Z. Rocek, Larval development in Oligocene palaeobatrachid frogs. Acta Paleontol. Pol. 48(4) (2003), 595-607.

[117] O.E. Roessler, Different types of chaos in two simple differential equations. Zeitschrift für Naturforsch A 31 (1976), 1664-1670.

[118] B. Sautois, S.R. Soffe, W.-C. Li and Alan Roberts, Role of type-specific neuron properties in a spinal cord motor network. To appear in J. Comp. Neurosci. (2007).

[119] K.T. Sillar and Alan Roberts, A neuronal mechanism for sensory gating during locomotion in a vertebrate. Nature 331(6153) (1988), 262-265.

[120] K.T. Sillar and Alan Roberts, Characterisation and function of spinal excitatory interneurons with commissural projections in Xenopus laevis embryos. Europ. J. Neurosci. 2 (1990), 1051-1062.

[121] S. Soffe and Alan Roberts, Activity of myotomal motoneurons during fictive swimming in frog embryos. J. Neurophysiol. 48 (1982), 1274-1288.

[122] S.R. Soffe, Roles of Glycinergic Inhibition and N-Methyl-D-Aspartate Receptor Mediated Excitation in the Locomotor Rhythmicity of One Half of the Xenopus Embryo Central Nervous System. Eur. J. Neurosci. 1 (1989), 561-571.

[123] S.R. Soffe, Two distinct rhythmic motor patterns are driven by common premotor and motor neurons in a simple vertebrate spinal cord. J. Neurosci. 13 (1993), 4456-4469.

[124] S.R. Soffe, F.Y. Zhao and Alan Roberts, Functional projection distances of spinal interneurons mediating reciprocal inhibition during swimming in Xenopus tadpoles. Eur. J. Neurosci. 13 (2001), 617-627.

[125] R. Tinsley, L. Minter, J. Measey, K. Howell, A. Veloso and H. Nez, Xenopus laevis. In: IUCN 2006, 2006 IUCN Red List of Threatened Species, www.iucnredlist.org (2006).

[126] R.D. Traub and R. Miles, Neuronal networks of the hippocampus. Cambridge University Press, New York (1991).

[127] K. Tsumoto, T. Yoshinaga, K. Aihara and H. Kawakami, Bifurcations in Morris-Lecar neuron model. Proceedings of the Int. Symp. on Nonlinear Theory and its Appls., Xi’an, China (2002), 363-366.

[128] M.J. Tunstall, Alan Roberts and S.R. Soffe, Modelling inter-segmental coordination of neuronal oscillators: synaptic mechanisms for uni-directional coupling during swimming in Xenopus tadpoles. J. Comput. Neurosci. 13 (2002), 143-158.

[129] C. van Vreeswijk and D. Hansel, Patterns of synchrony in neural networks with spike adaptation. Neural Comput. 13 (2001), 959-992.

[130] E. Wolf, F.Y. Zhao and Alan Roberts, Non-linear summation of excitatory synaptic inputs to small neurones: a case study in spinal motoneurones of the young Xenopus tadpole. J. Physiol. 511(3) (1998), 871-886.

Dynamische Systemen en hun Toepassingen in de Neurowetenschap

Nederlandse samenvatting

In dynamische systemen wordt de evolutie van een systeem voorgeschreven door de huidige toestand en door een set regels die voorschrijven hoe het in de toekomst zal evolueren. Door deze regels of vergelijkingen te volgen, kan men het gedrag van het systeem, in principe zo lang men wil, voorspellen. Deze heel algemene definitie laat toe dat wetenschappers uit een groot aantal wetenschapsgebieden dynamische systemen gebruiken bij hun onderzoek. Dynamische systemen kunnen inderdaad gebruikt worden voor het modelleren van elk proces, zolang de evolutie ervan maar volledig kan omschreven worden in een eindig aantal regels of vergelijkingen. Enkele van de wetenschappelijke domeinen waarin dynamische systemen gebruikt worden voor het modelleren van toepassingen zijn fysica, economie, geologie, biologie en chemie.

Door het grote aantal mogelijke toepassingen is er ook veel interesse in de algemene theorie achter dynamische systemen. De wijziging van (de toestand van) een systeem door verandering van een parameter kan vaak fenomenen verklaren die worden waargenomen in natuur en wetenschap. Een radicale wijziging in het gedrag van een systeem bij een bepaalde parameterwaarde wordt een bifurcatie genoemd; de studie van deze bifurcaties (zgn. bifurcatietheorie) is een heel belangrijk onderzoeksdomein in de wiskunde. In Hoofdstuk 1 van deze doctoraatsthesis wordt een inleiding gegeven tot continue (in tegenstelling tot discrete) dynamische systemen, en de belangrijkste bifurcaties worden hierbij besproken. Dit hoofdstuk zou alle achtergrondinformatie moeten bevatten over dynamische systemen die de lezer moet hebben om de rest van deze thesis te kunnen begrijpen.

Een belangrijk domein van toepassing van dynamische systemen is computationele neurobiologie. Hier worden dynamische systemen geregeld gebruikt voor het modelleren van zenuwcellen en/of zenuwnetwerken, waarbij de mate van detail van de modellen kan variëren. Het eerste neurale model dat gebaseerd was op individuele ionenstromen in en uit een zenuwcel werd ontwikkeld door Hodgkin en Huxley in 1952. Sindsdien zijn een groot aantal modellen geconstrueerd, in meerdere of mindere mate gebaseerd op de ideeën achter dit oorspronkelijke model. In Hoofdstuk 2 wordt een uitgebreide introductie gegeven tot de neurobiologische achtergrond, met o.a. uitleg over celmembraan, ionenstromen en ionenkanalen, gevolgd door een opsomming van vier bekende neurale modellen, die opgebouwd zijn aan de hand van dynamische systemen. Deze zelfde modellen zullen in verdere delen van deze thesis nog terugkeren als typevoorbeelden bij verschillende toepassingen.

Het is een haast onmogelijke taak om alle bestaande neurale modellen bij te houden en daardoor is de studie van deze modellen een onderzoeksgebied op zichzelf. Een belangrijk concept dat hierbij kan helpen is de classificatie van de ontwikkelde modellen. Als er een rigoureuze classificatie zou bestaan, waarin elke categorie een specifieke set eigenschappen heeft, en waarbij er duidelijke criteria zouden bestaan om modellen in één van deze categorieën onder te brengen, dan zou de studie van nieuw ontwikkelde modellen sterk vereenvoudigd worden. Dit zou dan nog vooral bestaan uit het bepalen tot welke categorie het nieuwe model behoort, waarna je onmiddellijk een waslijst van belangrijke eigenschappen en/of functionaliteiten van het model zou kennen. Er bestaan al een aantal aanzetten tot dergelijke classificaties, maar ze vertonen elk nog duidelijke gebreken op een aantal vlakken. In Sectie 2.3 verduidelijken we een aantal van deze gebreken en suggereren we een verbeterde versie.

Verder introduceren we in Hoofdstuk 2 de zogenaamde ’Phase Response Curve’ of PRC. Deze curve is een van oorsprong experimenteel gedefinieerd concept, dat ook wiskundig kan beschouwd worden en dat heel belangrijke toepassingen heeft in het gebied van de neurobiologie. Het is een curve die het effect beschrijft dat een inkomend signaal heeft op het gedrag van een periodieke baan. Aangezien een actief piekend neuron kan voorgesteld worden als zo’n periodieke baan, kan de PRC gebruikt worden om de impact aan te duiden van een inkomende impuls op een zenuwcel (bijvoorbeeld van een naburige cel). In dit hoofdstuk leiden we een nieuwe en efficiënte methode af om deze PRC van een periodieke baan te berekenen.

Doordat de studie van dynamische systemen en bifurcaties wijd verspreide interesse opwekt bij een wetenschappelijk publiek, zijn in de loop der decennia verscheidene softwarepakketten ontwikkeld om the helpen bij die studie. Belangrijke en veelgebruikte voorbeelden daarvan zijn AUTO en CONTENT. Alle ontwikkelde pakketten vertonen echter een aantal niet onbelangrijke nadelen. AUTO is snel en efficiënt, maar het heeft geen gebruikersinterface en is, deels daardoor, niet gebruiksvriendelijk. Het vraagt redelijk wat tijd en moeite van de doorsnee wetenschapper om de software voldoende te begrijpen om grondige studies te kunnen uitvoeren op gewenste dynamische systemen. CONTENT is veel gebruiksvriendelijker, maar moeilijk te installeren, en recent is de verdere ontwikkeling en opvolging van het pakket verwaterd. Een ander belangrijk nadeel van beide pakketten is de platformafhankelijkheid. Beide zijn oorspronkelijk ontwikkeld voor Unix / Linux computers, terwijl de meeste wetenschappers tegenwoordig veel beter vertrouwd zijn met het MS Windows besturingssysteem (tot spijt van wie het benijdt). En hoewel er een versie van CONTENT ontwikkeld werd voor Windows, vormt de moeilijkheid van installatie een belangrijke drempel.

In het jaar 2000 werd begonnen met de ontwikkeling van MATCONT, een nieuw softwarepakket voor de studie van dynamische systemen en hun bifurcaties, met de bedoeling om de belangrijkste problemen van bestaande software op te lossen. Het was volledig geschreven in MATLAB, omdat dit wiskundig softwarepakket door veel wetenschappers gebruikt wordt, omdat het platformonafhankelijkheid ondersteunt en om de uitbreidbaarheid van het pakket te vergroten. De gebruikersinterface was gebaseerd op die van CONTENT, om de gebruiksvriendelijkheid van dat pakket over te nemen en om het gemakkelijker te maken voor gebruikers om over te stappen van het ene pakket naar het andere.

In Hoofdstuk 3 worden een aantal belangrijke bijdragen tot dit softwarepakket uit de doeken gedaan. Zo werden de functionaliteiten van het programma uitgebreid met de mogelijkheid om de PRC van een periodieke baan te laten berekenen. Ook de afgeleide van de PRC kan op deze manier bekomen worden. De implementatie hiervan is gebaseerd op het wiskundig algoritme uit Hoofdstuk 2, maar bovendien maakt ze zeer efficiënt gebruik van het specifieke continuatiegedeelte van het softwarepakket MATCONT.

De gebruiker kan nu ook homoclinische bifurcaties detecteren en continueren. Deze curven van homoclinische banen kunnen starten van een Bogdanov-Takens bifurcatie of van een periodieke baan met grote periode. Alle bekende bifurcaties langs deze curven kunnen gedetecteerd worden.

Tenslotte hebben we de snelheid van het softwarepakket aanzienlijk verhoogd door C-code te introduceren tussen de MATLAB-routines. Door nuttig gebruik te maken van deze C-code zijn we erin geslaagd om de snelheid van continuaties van periodieke banen te verhogen met meer dan de helft en dit alles zonder het gemak van installatie en de platformonafhankelijkheid te verliezen die zo typisch waren voor MATCONT. In een laatste deel van dit hoofdstuk rapporteren we een belangrijke programmeerfout die we hebben ontdekt in de ingebouwde functionaliteiten van MATLAB voor ijle matrices.

MATCONT werd ontwikkeld als een wiskundig geöriënteerd softwarepakket, vooral gericht op de theoretische studie van dynamische systemen. Tegenwoordig wordt het veel gebruikt in onderwijs en wiskundig onderzoek. De mogelijke toepassingen zijn echter allesbehalve beperkt tot die specifieke domeinen. MATCONT kan gebruikt worden in elk wetenschappelijk gebied waarbij dynamische systemen gebruikt worden. De theoretische studie van deze systemen kan bijdragen tot het verklaren van bepaalde fenomenen. Dat MATCONT interessante toepassingen heeft in bijvoorbeeld neurobiologie, wordt geïllustreerd in Hoofdstuk 4.

Een eerste logisch gebruik van de software is het bestuderen van het bifurcatiediagram van een bestaand neuraal model. Het Morris-Lecar model is zo'n wijd verspreid en veel gebruikt model, maar tot nog toe was er nog door niemand een complete bifurcatieanalyse gemaakt van dit systeem. Dit type van uitgebreid bifurcatiediagram kan belangrijk zijn om te weten welk gedrag kan voorkomen bij bepaalde parameterwaarden in het systeem. Op deze manier kan men bepaald waargenomen gedrag in een netwerk gaan verklaren, of een specifiek gewenst gedrag veroorzaken. We hebben als eerste zo'n compleet bifurcatiediagram van die bekende model opgebouwd.

We introduceren ook verscheidene interessante toepassingen van PRCs, die nu een integraal deel van MATCONT uitmaken. We tonen hoe deze curven kunnen gebruikt worden om de mogelijkheden te onderzoeken van neurale netwerken om te synchroniseren of in fase te vuren en hoe de vertraging van de verbinding tussen zenuwcellen hierin een cruciale rol speelt. We illustreren ook een aantal interessante fenomenen in de interactie van PRCs en bifurcaties van periodieke banen.

Dynamische systemen zijn niet alleen nuttig in theoretische neurobiologie. Ook in toegepaste neurobiologie hebben ze een belangrijke rol te vervullen. Wanneer dynamische systemen gebruikt worden om fysiologische neuronen te modelleren, kunnen de resulterende modellen gebruikt worden om de interactie van deze neuronen te bestuderen in een biologisch relevant opgebouwd netwerkmodel.

In samenwerking met een groep biologen aan de University of Bristol (in het Verenigd Koninkrijk) hebben we gedetailleerde modellen opgebouwd van de zenuwcellen in de ruggengraat van het pasgeboren kikkervisje van de *Xenopus laevis*. De biologische achtergrond over dit diertje en de vergelijkingen en parameters die worden gebruikt voor de modellen van de individuele neuronen en verbindingen worden gegeven in Hoofdstuk 5.

In Hoofdstuk 6 richten we ons op het oorspronkelijke doel van deze samenwerking. De individuele modellen van zenuwcellen en synapsen worden gebruikt om biologisch realistische netwerken van neuronen op te bouwen, zoals ze kunnen gevonden worden in de ruggengraat van de jonge *Xenopus*.

Het eerste netwerk dat we bouwden was bedoeld om het zwemgedrag van het kikkervisje te simuleren. De effectieve interacties binnen dit netwerk waren reeds voor het merendeel bekend en dit netwerk werd in de eerste plaats gebruikt om onze modellen voor de individuele neuronen en synapsen te verifiëren, maar ook om het belang aan te tonen van specificiteit in neurale modellering. We tonen aan dat het negeren van een aantal belangrijke verschillen die we hebben ingebouwd in de modellen voor de verschillende neuronen, resulteert in het verlies van de goede werking van het netwerk.

Hierna hebben we deze zelfde individuele modellen gebruikt om een andere hypothese te bestuderen, een hypothese op het vlak van synaptogenese. In het fysiologisch netwerk is het voor de meeste types neuronen waarschijnlijker om synaptische verbindingen te maken met specifieke types neuronen dan met andere types. De hypothese is dat deze specificiteit in verbinding puur gebaseerd is op de anatomische organisatie van de neuronen en niet op de mogelijkheid van groeiende synapsen om een onderscheid te maken tussen verschillende types zenuwcellen. Het feit dat een neuron voornamelijk contact zou leggen met één of meer specifieke types zenuwcellen, zou enkel veroorzaakt worden door de dorsoventrale positie van elk van die types zenuwcellen en de plaats waar die synapsen precies groeien. We hebben deze hypothese getest door netwerken op te bouwen bestaande uit de zwemneuronen, waarbij synapsen werden gecreëerd op puur probabilistische basis. De modelleringresultaten, die onze hypothese sterk ondersteunen, worden besproken in het tweede deel van Hoofdstuk 6.

Tenslotte hebben we een eerste versie ontwikkeld van een netwerk om het tweede soort beweging van het kikkervisje te modelleren, namelijk worstelen. Tijdens dit proces worden korte reeksen van pieken afgevuurd afwisselend aan beide zijden van het lichaam van het kikkervisje (in tegenstelling tot bij zwemmen, waarbij het gaat om afwisselende individuele pieken aan weerszijden). Van het neuraal netwerk dat actief is bij het worstelen, en van het gedrag op zich, is veel minder geweten dan van het zwemmen. Maar ons modelleren heeft onder andere aangetoond dat het meest waarschijnlijke mechanisme dat een reeks van pieken doet ophouden, synaptische plasticiteit zou zijn: door aan hoge frequentie signalen over één enkele synaps te sturen naar een ander neuron, zal een synaps verzwakken en uiteindelijk zo verzwakt raken dat het tijdelijk zal stoppen het signaal door te geven, tot het hersteld is. Door dit verschijnsel in te bouwen in ons model, waren wij in staat relatief betrouwbaar alternerende reeksen van pieken uit te lokken aan beide zijden van het lichaam van de *Xenopus*.

De resultaten in deze doctoraatsthesis werden gepubliceerd in, of ingezonden voor publicatie naar, gespecialiseerde wetenschappelijke tijdschriften, zie [33], [49], [57], [58], [59], [60], [118], [35].