

Switching to Nonhyperbolic Cycles from Codimension Two Bifurcations
of Equilibria of Delay Differential Equations

Non Peer-reviewed author version

BOSSCHAERT, Maikel; Janssens, Sebastiaan G. & Kuznetsov, Yu. A. (2020)
Switching to Nonhyperbolic Cycles from Codimension Two Bifurcations of Equilibria
of Delay Differential Equations. In: SIAM journal on applied dynamical systems, 19
(1) , p. 252 -303.

DOI: 10.1137/19M1243993

Handle: <http://hdl.handle.net/1942/34534>

Switching to nonhyperbolic cycles from codimension two bifurcations of equilibria of delay differential equations

M.M. Bosschaert* S.G. Janssens† Yu.A. Kuznetsov‡

March 19, 2019

Abstract: In this paper we perform the parameter-dependent center manifold reduction near the generalized Hopf (Bautin), fold-Hopf, Hopf-Hopf and transcritical-Hopf bifurcations in delay differential equations (DDEs). This allows us to initialize the continuation of codimension one equilibria and cycle bifurcations emanating from these codimension two bifurcation points. The normal form coefficients are derived in the functional analytic perturbation framework for dual semigroups (sun-star calculus) using a normalization technique based on the Fredholm alternative. The obtained expressions give explicit formulas which have been implemented in the freely available numerical software package `DDE-BifTool`. While our theoretical results are proven to apply more generally, the software implementation and examples focus on DDEs with finitely many discrete delays. Together with the continuation capabilities of `DDE-BifTool`, this provides a powerful tool to study the dynamics near equilibria of such DDEs. The effectiveness is demonstrated on various models.

Keywords: Generalized Hopf (Bautin) bifurcation, fold-Hopf bifurcation, Hopf-Hopf bifurcation, transcritical-Hopf bifurcation, codimension two bifurcation, normal forms, nonhyperbolic cycles, branch switching, delay differential equations, Center Manifold Theorem, adjoint operator semigroups, sun-star calculus, `DDE-BifTool`

2010 MSC: 34K18, 34K19, 34K60, 37L10, 37M20

1 Introduction

Great interest has recently been shown in the analysis of degenerate Hopf bifurcations in delay differential equations (DDEs), see e.g. [1, 17, 18, 25, 35, 36, 39, 40, 48, 50, 51, 54, 57, 61, 64, 65, 66]. In the simplest case, often encountered in applications, such DDEs have the form

$$\dot{x}(t) = f(x(t), x(t - \tau_1), \dots, x(t - \tau_m), \alpha), \quad t \geq 0, \quad (1)$$

where $x(t) \in \mathbb{R}^n$, $\alpha \in \mathbb{R}^p$, $f : \mathbb{R}^{n \times (m+1)} \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is a smooth mapping and the delays $0 < \tau_1 < \dots < \tau_m$ are constant. They are known as *discrete* DDEs or DDEs of *point type*.

*Department of Mathematics, Hasselt University, Diepenbeek Campus, Agoralaan Gebouw D, 3590 Diepenbeek, Belgium (maikel.bosschaert@uhasselt.be).

†Department of Mathematics, Utrecht University, Budapestlaan 6, 3508 TA Utrecht, The Netherlands (s.g.janssens@uu.nl, sj@dydx.nl).

‡Department of Mathematics, Utrecht University, Budapestlaan 6, 3508 TA Utrecht, The Netherlands and Department of Applied Mathematics, University of Twente, Zilverling Building, 7500AE Enschede, The Netherlands (I.A.Kouznetsov@uu.nl).

Using the framework of perturbation theory for dual semigroups developed in [4,5,6,7] the existence of a finite dimensional smooth center manifold for DDEs can be rigorously established [14]. As a consequence the normalization method for local bifurcations of ODEs developed in [42] can be lifted [37] rather easily to the infinite dimensional setting of DDEs. One of the advantages of this normalization technique is that the center manifold reduction and the calculation of the normal form coefficients are performed simultaneously by solving the so-called *homological equation*. The method gives explicit expressions for the coefficients rather than a procedure as developed in [22,23]. The critical normal form coefficients for all five generic codimension two bifurcations of equilibria of DDEs have been derived [37] and partially implemented [60] into the fully GNU Octave compatible MATLAB package DDE-BifTool [21,56].

In this paper we will perform the parameter-dependent center manifold reduction and normalization for three codimension two Hopf cases: the *generalized Hopf*, *fold-Hopf* and *Hopf-Hopf* bifurcations. This will allow us to initialize the continuation of codimension one bifurcation curves of nonhyperbolic equilibria and cycles emanating from the codimension two points. These are the only codimension two bifurcation points of equilibria in generic DDEs where codimension one bifurcation curves of nonhyperbolic cycles could originate. We also treat the more special transcritical-Hopf bifurcation which is frequently found in applications.

The center manifold theorem for parameter-dependent DDEs as presented in [14] assumes explicitly that the steady state exists for all nearby parameter values. However, for a generic fold-Hopf bifurcation this assumption is not satisfied. An attempt to deal with this complication has been made in [30], where it is discussed how to reduce a parameter-dependent DDE to a DDE without parameters by appending the trivial equation $\dot{\alpha} = 0$. However, the reduction in [30] is based on the formal adjoint approach [32] and applies specifically to DDEs, while at times it lacks consistency. Therefore we demonstrate in this paper how the reduction to the parameter-independent case can be done in the sun-star framework, enabling a rigorous approach to the existence of parameter-dependent center manifolds for a class of evolution equations that includes DDEs. This allows us to treat bifurcations of equilibria with zero eigenvalues in generic DDEs while at the same time achieving applicability of our results to other classes of delay equations.

This paper is organized as follows. In **Section 2** we offer a concise review of perturbation theory for dual semigroups (also called sun-star calculus), both on an abstract level as well as in application to the analysis of classical DDEs as dynamical systems. We also recall from [37] various results that are needed for the normalization.

In **Section 3** we show how the theory from the previous section also applies to parameter-dependent classical DDEs by converting them into a parameter-*independent* system on a product state space. We again present the material in two stages: Results are first established at a more abstract semigroup level and next applied to classical DDEs depending on parameters. In particular, we define the parameter-dependent local center manifold and give an explicit ODE for solutions that are confined to it.

In **Section 4** we describe the general technique used to derive expressions for the normal form coefficients on the parameter-dependent center manifold in the infinite dimensional setting of classical DDEs.

Before we apply this technique to the previously mentioned codimension two bifurcations, we summarize in **Section 5** relevant smooth normal forms and we list asymptotics for the codimension one cycle bifurcation curves emanating from the codimension two points as well as for the corresponding nonhyperbolic equilibria.

In **Section 6** the method is then applied to the generalized Hopf (Bautin), fold-Hopf, Hopf-Hopf and transcritical-Hopf bifurcations in classical DDEs. We provide explicit expressions for all normal form coefficients necessary for the predictors of codimension one bifurcation curves. While most of the critical normal form coefficients for these bifurcations were obtained in [37], we briefly re-derive them to ensure readability.

In **Section 7** we provide explicit computational formulas for the evaluation of the linear and multi-linear forms used in the normal form coefficients and predictors for the simplest subclass consisting of

DDEs (1) with finite many discrete delays. These formulas are actually implemented in `DDE-BifTool`.

In [Section 8](#) we employ our implementation in `DDE-BifTool` to illustrate the accuracy of the codimension one bifurcation curve predictors through various example models displaying all aforementioned degenerate Hopf cases. A complete step-by-step walk-through of the examples, including all code to reproduce the obtained results, is provided in the [Supplement](#).

2 Dual perturbation theory and classical DDEs

We begin by presenting those general elements of perturbation theory for dual semigroups that are useful for the study of classical DDEs as dynamical systems. Throughout we assume sun-reflexivity - a term that will be introduced in [Section 2.1](#). From [Section 2.4](#) onward, we then explain how the general results apply to classical DDEs. The standard reference for this entire section is [\[14\]](#), while for the underlying theory of semigroups of linear operators we recommend [\[19, 20\]](#).

2.1 Duality structure and linear perturbation

The starting point is a \mathcal{C}_0 -semigroup T_0 on a real or complex Banach space X . Let A_0 with domain $\mathcal{D}(A_0)$ be the infinitesimal generator (or: generator, for short) of T_0 . We denote by X^* the topological dual space of X and we use the prefix notation for the pairing between $x^* \in X^*$ and $x \in X$,

$$\langle x^*, x \rangle := x^*(x).$$

If X is not reflexive then the adjoint semigroup T_0^* is in general only weak* continuous on X^* and A_0^* generates T_0^* only in the weak* sense. The maximal subspace of strong continuity

$$X^\odot := \{x^* \in X^* : t \mapsto T_0^*(t)x^* \text{ is norm-continuous on } \mathbb{R}_+\}$$

is invariant under T_0^* and we have the characterization

$$X^\odot = \overline{\mathcal{D}(A_0^*)}$$

where the bar denotes the norm closure in X^* . By construction the restriction of T_0^* to X^\odot is a \mathcal{C}_0 -semigroup that we denote by T_0^\odot . Its generator A_0^\odot is the *part* of A_0^* in X^\odot ,

$$\mathcal{D}(A_0^\odot) = \{x^\odot \in \mathcal{D}(A_0^*) : A_0^*x^\odot \in X^\odot\}, \quad A_0^\odot x^\odot = A_0^*x^\odot.$$

At this stage we again have a \mathcal{C}_0 -semigroup T_0^\odot with generator A_0^\odot on a Banach space X^\odot so we can iterate the above construction. On the dual space $X^{\odot*}$ we obtain the adjoint semigroup $T_0^{\odot*}$ with weak* generator $A_0^{\odot*}$. By restriction to the maximal subspace of strong continuity $X^{\odot\odot} = \overline{\mathcal{D}(A_0^{\odot*})}$ we end up with the \mathcal{C}_0 -semigroup $T_0^{\odot\odot}$. Its generator $A_0^{\odot\odot}$ is the part of $A_0^{\odot*}$ in $X^{\odot\odot}$.

The canonical injection $j : X \rightarrow X^{\odot*}$ defined by

$$\langle jx, x^\odot \rangle := \langle x^\odot, x \rangle \tag{2}$$

maps X into $X^{\odot\odot}$. If j maps X onto $X^{\odot\odot}$ then X is called \odot -*reflexive* (pronounce: sun-reflexive) with respect to T_0 . One may define an equivalent norm on X with respect to which j becomes an isometry, but this need not be assumed. However, *sun-reflexivity of X with respect to T_0 will be assumed throughout*.

With the abstract duality structure in place, we next turn our attention to perturbation. Let $L : X \rightarrow X^{\odot*}$ be a bounded linear operator. Then there exists a unique \mathcal{C}_0 -semigroup T on X that satisfies the linear integral equation

$$T(t)x = T_0(t)x + j^{-1} \int_0^t T_0^{\odot*}(t-\tau)LT(\tau)x \, d\tau, \quad t \geq 0, x \in X, \tag{3}$$

where the weak* Riemann integral takes values in $X^{\odot\odot}$ and the running assumption of sun-reflexivity justifies the application of j^{-1} . By using (3) to express the difference $T - T_0$ of the perturbed and the unperturbed semigroups, one proves that the maximal subspaces of strong continuity X^\odot and $X^{\odot\odot}$ are *the same* for T and T_0 , so there is no need to distinguish them with a subscript. In particular, X is sun-reflexive also with respect to T . On $X^{\odot*}$ the perturbation L appears additively in the action of $A^{\odot*}$,

$$\mathcal{D}(A^{\odot*}) = \mathcal{D}(A_0^{\odot*}), \quad A^{\odot*} = A_0^{\odot*} + Lj^{-1}. \quad (4)$$

We recover the generator A of T by considering the part of $A^{\odot*}$ in $X^{\odot\odot}$. As a consequence L moves into the domain and we find

$$\mathcal{D}(A) = \{x \in X : jx \in \mathcal{D}(A_0^{\odot*}) \text{ and } A_0^{\odot*}jx + Lx \in X^{\odot\odot}\}, \quad Ax = j^{-1}(A_0^{\odot*}jx + Lx).$$

For proofs of the statements so far, see [14, Appendix II.3 and Chapter III].

2.2 Nonlinear perturbation and linearization

The \mathcal{C}_0 -semigroup T arose as a linear perturbation of the original \mathcal{C}_0 -semigroup T_0 , so the next step is to introduce a nonlinear perturbation of T itself. In keeping with the tradition for nonlinear problems [14, Sections VII.1 and VIII.1] we only regard the case that X is a *real* Banach space, also see **Remark 1** below. Let $R : X \rightarrow X^{\odot*}$ be a C^k -operator for some $k \geq 1$ such that

$$R(0) = 0, \quad DR(0) = 0,$$

and consider the nonlinear integral equation

$$u(t) = T(t)x + j^{-1} \int_0^t T^{\odot*}(t - \tau)R(u(\tau)) d\tau, \quad t \geq 0, x \in X. \quad (5)$$

Due to the nonlinearity, for a given initial condition $x \in X$ one can at most guarantee existence of a *maximal solution* $u_x : I_x \rightarrow X$ of (5) on a forward time interval $I_x := [0, t_x)$ for some $0 < t_x \leq \infty$ [14, Chapter VII]. The family of all such maximal solutions defines a nonlinear semiflow $\Sigma : \mathcal{D}(\Sigma) \rightarrow X$,

$$\mathcal{D}(\Sigma) := \{(t, x) \in [0, \infty) \times X : t \in I_x\}, \quad \Sigma(t, x) := u_x(t), \quad (6)$$

that may in addition depend on parameters [14, Defs. VII.2.1 and VII.2.9]. (For reasons discussed in **Section 3**, we will treat parameter dependence differently and separately. Until then, the reader can consider all parameters to be held fixed and absent in the notation.) The domain of Σ is open in $[0, \infty) \times X$ and $0 \in X$ is a stationary point of Σ ,

$$I_0 = [0, \infty), \quad \Sigma(t, 0) = 0, \quad \forall t \geq 0.$$

The semiflow Σ is (in fact, uniformly) differentiable with respect to the state at $(t, 0) \in \mathcal{D}(\Sigma)$, with the partial derivative

$$D_2\Sigma(t, 0) = T(t), \quad \forall t \geq 0, \quad (7)$$

where T is the \mathcal{C}_0 -semigroup that satisfies (3).

Remark 1. For nonlinear problems it is customary to work on a real Banach space X . The reason is that these problems often come from concrete equations with nonlinear right-hand sides for which it is unclear if and how they can be extended to complex arguments. Consequently, if we want to analyze the linearization of Σ at $0 \in X$ using spectral theory, then it becomes necessary to *complexify* X and the linear operators acting on X [14, Section III.7 and last part of Section IV.2], [52, Section 1.3]. In particular, by the spectrum of A we mean the spectrum of its complexification on the complexified Banach space.

2.3 Critical local center manifolds

As in [Section 2.2](#) we continue to assume that T_0 is a \mathcal{C}_0 -semigroup on a *real* Banach space X that is sun-reflexive with respect to T_0 . In addition we assume that T_0 is eventually compact and L is a compact operator. This implies that the perturbed semigroup T defined by [\(3\)](#) is eventually compact as well [[11](#), Theorem 2.8].

When considering solutions that exist for all (positive and negative) time - such as periodic orbits - it is useful to write [\(5\)](#) in the translation invariant form

$$u(t) = T(t-s)u(s) + j^{-1} \int_s^t T^{\odot*}(t-\tau)R(u(\tau)) d\tau, \quad -\infty < s \leq t < \infty. \quad (8)$$

A *solution* of [\(8\)](#) is a continuous function $u : I \rightarrow X$ on some nondegenerate - possibly unbounded - interval $I \subseteq \mathbb{R}$ that satisfies [\(8\)](#) for all $s, t \in I$ with $s \leq t$. Naturally, u is a solution of [\(8\)](#) if and only if

$$t-s \in I_{u(s)}, \quad u(t) = \Sigma(t-s, u(s)), \quad \forall s, t \in I \text{ with } s \leq t,$$

where $\Sigma : \mathcal{D}(\Sigma) \rightarrow X$ is the nonlinear semiflow from [\(6\)](#). The interval I is often left implicit.

The general center manifold theorems from [[14](#), Chapter IX] for equations of the type [\(8\)](#) apply to the particular case where T is an eventually compact \mathcal{C}_0 -semigroup on a real, sun-reflexive Banach space. Let us therefore suppose that $0 \in X$ is a nonhyperbolic equilibrium of Σ , so the generator A of T possesses $1 \leq n_0 < \infty$ purely imaginary eigenvalues, counting algebraic multiplicities - see [Remark 1](#). Let $X_0 \subseteq X$ be the *real* center eigenspace corresponding to these eigenvalues. Then there exists a C^k -smooth n_0 -dimensional *local* center manifold $\mathcal{W}_{\text{loc}}^c$ that is tangent to X_0 at the origin. Any solution $u : I \rightarrow X$ of [\(8\)](#) that lies on $\mathcal{W}_{\text{loc}}^c$ is differentiable on I and satisfies

$$j\dot{u}(t) = A^{\odot*}ju(t) + R(u(t)), \quad \forall t \in I, \quad (9)$$

where $A^{\odot*}$ is the weak* generator of $T^{\odot*}$. We note that [\(9\)](#) is an identity in $X^{\odot*}$.

2.4 The special case of classical DDEs

It will now be explained how the general results from [Sections 2.1](#) to [2.3](#) apply to classical DDEs. We choose the nonreflexive Banach space $X := C([-h, 0], \mathbb{R}^n)$ as the state space, introduce a C^k -smooth operator $F : X \rightarrow \mathbb{R}^n$, and consider an equation with a finite delay $0 < h < \infty$ of the form

$$\dot{x}(t) = F(x_t), \quad t \geq 0, \quad (\text{DDE})$$

with an initial condition

$$x_0 = \varphi \in X. \quad (\text{IC})$$

For each $t \geq 0$, the function $x_t : [-h, 0] \rightarrow \mathbb{R}^n$ defined by

$$x_t(\theta) := x(t+\theta), \quad \forall \theta \in [-h, 0],$$

is called the *history* of the unknown function x at time t . Equations of the type [\(DDE\)](#) will be called *classical* DDEs. Note that [\(1\)](#) is quite literally a case in point. By a *solution of the initial value problem* [\(DDE\)](#)–[\(IC\)](#) we mean a continuous function $x : [-h, t_+) \rightarrow \mathbb{R}^n$ for some $0 < t_+ \leq \infty$ that is differentiable on $[0, t_+)$ and satisfies [\(DDE\)](#) and [\(IC\)](#). When $t_+ = \infty$ we call x a *global solution*.

We want to study [\(DDE\)](#) near an equilibrium at the origin, so assume that $F(0) = 0$ and split F into its linear and nonlinear parts,

$$F(\varphi) = \int_0^h d\zeta(\theta)\varphi(-\theta) + G(\varphi), \quad \varphi \in X.$$

Here $\zeta : [0, h] \rightarrow \mathbb{R}^{n \times n}$ is a matrix-valued function of bounded variation, normalized by the requirement that $\zeta(0) = 0$ and ζ is right-continuous on the open interval $(0, h)$. The integral is of the Riemann-Stieltjes type, and $G : X \rightarrow \mathbb{R}^n$ is a C^k -smooth nonlinear operator. It is common to denote the linear part more succinctly as

$$\langle \zeta, \varphi \rangle := \int_0^h d\zeta(\theta) \varphi(-\theta), \quad (10)$$

so that

$$F(\varphi) = \langle \zeta, \varphi \rangle + G(\varphi), \quad \varphi \in X. \quad (11)$$

We first consider the case $G = 0$, whence (DDE) reduces to the linear equation

$$\dot{x}(t) = \langle \zeta, x_t \rangle, \quad t \geq 0. \quad (12)$$

In order to understand the relationship between (12) and (3) we begin by observing that the trivial DDE

$$\dot{x}(t) = 0, \quad t \geq 0, \quad (13)$$

with initial condition (IC) has the obvious solution

$$x^\varphi(t) = \begin{cases} \varphi(t), & t \in [-h, 0], \\ \varphi(0), & t > 0. \end{cases}$$

Using this solution, we define the strongly continuous *shift semigroup* T_0 on X by

$$(T_0(t)\varphi)(\theta) := x^\varphi(t + \theta) = \begin{cases} \varphi(t + \theta), & t + \theta \in [-h, 0], \\ \varphi(0), & t + \theta > 0. \end{cases} \quad (14)$$

We note that $T_0(h)$ is a compact operator, so T_0 is eventually compact. For this particular combination of X and T_0 the abstract duality structure from Section 2.1 can be constructed systematically and explicitly [14, Section II.5]. We only summarize the few facts that will be used in the sequel.

Remark 2 (Notation). For $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ let \mathbb{K}^n be the linear space of column vectors and let \mathbb{K}^{n*} be the linear space of row vectors, both over \mathbb{K} . Elements of \mathbb{K}^n are denoted by $q = (q_1, q_2, \dots, q_n)$ - commas between the entries - while elements in \mathbb{K}^{n*} are denoted by $p = (p_1 p_2 \cdots p_n)$ - no commas between the entries. We sometimes use the pairing defined by the row-column matrix multiplication:

$$p \cdot q := pq = \sum_{j=1}^n p_j q_j, \quad p \in \mathbb{K}^{n*}, q \in \mathbb{K}^n.$$

Note that the standard Hermitian inner product between two vectors $p^T, q \in \mathbb{C}^n$ should be written as $\bar{p} \cdot q$ and *not* as $p \cdot q$.

On X^\odot : The maximal domain of strong continuity of T_0^* has the representation

$$X^\odot = \mathbb{R}^{n*} \times L^1([0, h], \mathbb{R}^{n*}), \quad (15)$$

and the duality pairing between $\varphi^\odot = (c, g) \in X^\odot$ and $\varphi \in X$ is

$$\langle \varphi^\odot, \varphi \rangle = c\varphi(0) + \int_0^h g(\theta) \varphi(-\theta) d\theta. \quad (16)$$

On $X^{\odot*}$: Switching to the dual space of (15) yields the representation

$$X^{\odot*} = \mathbb{R}^n \times L^\infty([-h, 0], \mathbb{R}^n),$$

and the duality pairing between $\varphi^{\odot*} = (a, \psi) \in X^{\odot*}$ and $\varphi^\odot = (c, g) \in X^\odot$ is

$$\langle \varphi^{\odot*}, \varphi^\odot \rangle = ca + \int_0^h g(\theta) \psi(-\theta) d\theta. \quad (17)$$

The canonical injection (2) sends $\varphi \in X$ to $j\varphi = (\varphi(0), \varphi)$, mapping X onto $X^{\odot\odot}$. Therefore X is sun-reflexive with respect to the shift semigroup T_0 .

Next, we specify the linear and nonlinear perturbations L and R in (3) and (5), respectively, and to relate these two abstract integral equations in X to the linear and nonlinear initial value problems for (DDE). For $i = 1, \dots, n$ we denote $r_i^{\odot*} := (e_i, 0)$ where e_i is the i th standard basis vector of \mathbb{R}^n . It is conventional and convenient to introduce the shorthand

$$wr^{\odot*} := \sum_{i=1}^n w_i r_i^{\odot*}, \quad \forall w = (w_1, \dots, w_n) \in \mathbb{R}^n,$$

and we note that $wr^{\odot*} = (w, 0) \in X^{\odot*}$. First we define the compact linear perturbation in (3) as

$$L\varphi := \langle \zeta, \varphi \rangle r^{\odot*}, \quad (18)$$

where the pairing in the right-hand side is given by (10). Now (12) with (IC) is equivalent to (3) with (18) in the following sense: If T is the unique \mathcal{C}_0 -semigroup on X satisfying (3) with (18) then $x^\varphi : [-h, \infty) \rightarrow \mathbb{R}^n$ defined by

$$x_0^\varphi := \varphi, \quad x^\varphi(t) := (T(t)\varphi)(0), \quad \forall t \geq 0,$$

is the unique global solution of (12) with (IC) and

$$x_t^\varphi = T(t)\varphi, \quad \forall t \geq 0.$$

It remains to specify the nonlinear perturbation R in (5) as

$$R(\varphi) := G(\varphi)r^{\odot*}, \quad (19)$$

where G is the nonlinear operator appearing in the splitting (11). Let Σ as in (6) be the nonlinear semiflow generated by the family of maximal solutions of (5) with (19). The equivalence between (DDE)–(IC) and (5) with (19) can be formulated as follows [14, Prop. VII.6.1]. The function $x^\varphi : [-h, t_\varphi) \rightarrow \mathbb{R}^n$ defined by

$$x_0^\varphi := \varphi, \quad x^\varphi(t) := \Sigma(t, \varphi)(0), \quad \forall t \in I_\varphi,$$

is the *maximal solution* of (DDE)–(IC), in the sense that any other solution necessarily exists only on a subinterval $[-h, t_+)$ for some $0 < t_+ \leq t_\varphi$ and coincides with x^φ there. Moreover,

$$x_t^\varphi = \Sigma(t, \varphi), \quad \forall t \in I_\varphi.$$

It is the content of (7) that generation and linearization commute: Starting with (DDE), linearization of the semiflow Σ at the stationary point $0 \in X$ yields precisely the eventually compact \mathcal{C}_0 -semigroup T corresponding to the linearized DDE (12).

2.5 Spectral computations for classical DDEs

The eventual compactness of T implies that the spectrum of its generator A - see [Remark 1](#) - consists entirely of isolated eigenvalues of finite algebraic multiplicity. It is clear from [\(18\)](#) that L is not just compact, but actually of finite rank. This implies that all spectral information about A is contained in a holomorphic *characteristic matrix function* $\Delta : \mathbb{C} \rightarrow \mathbb{C}^{n \times n}$ defined by

$$\Delta(z) := zI - \hat{\zeta}(z) \quad \text{with} \quad \hat{\zeta}(z) := \int_0^h e^{-z\theta} d\zeta(\theta), \quad (20)$$

where ζ is the *real kernel* from [\(10\)](#) [[14](#), Sections IV.4 and IV.5]. In particular, the eigenvalues of A are the roots of the *characteristic equation*

$$\det \Delta(z) = 0, \quad (21)$$

and the algebraic multiplicity of an eigenvalue equals its order as a root of [\(21\)](#).

We will be concerned exclusively with *simple* eigenvalues, for which the geometric and algebraic multiplicities are both equal to one. Let $\lambda \in \mathbb{C}$ be such a simple eigenvalue of A . There exist nonzero right and left null vectors $q \in \mathbb{C}^n$ and $p \in \mathbb{C}^{n*}$ of $\Delta(\lambda)$,

$$\Delta(\lambda)q = 0, \quad p\Delta(\lambda) = 0.$$

The second equation is of course equivalent to p^T being a nonzero right null vector of $\Delta^T(\lambda)$. The one-dimensional eigenspaces of A and A^* corresponding to λ are spanned by eigenfunctions φ and φ° , respectively, with

$$\varphi(\theta) = e^{\lambda\theta}q, \quad \theta \in [-h, 0], \quad (22)$$

and

$$\varphi^\circ = \left(p, \theta \mapsto p \int_\theta^h e^{\lambda(\theta-\tau)} d\zeta(\tau) \right), \quad \theta \in [0, h]. \quad (23)$$

We note that we have implicitly used - and will use consistently - the complexifications of X and of the representation [\(15\)](#) of X° . For a simple eigenvalue λ ,

$$\langle \varphi^\circ, \varphi \rangle \neq 0,$$

where the duality pairing is understood to be the complexification of [\(16\)](#). This nonequality implies that the eigenfunctions can be normalized to satisfy $\langle \varphi^\circ, \varphi \rangle = 1$. In fact, from [\(16\)](#) and [\(22\)](#) one computes

$$\langle \varphi^\circ, \varphi \rangle = p\Delta'(\lambda)q, \quad (24)$$

so this normalization can be effectuated by scaling p and q such that $p\Delta'(\lambda)q = 1$. Finally, it is easily seen that if $\mu \neq \lambda$ is another simple eigenvalue of A with eigenvector ψ and adjoint eigenvector ψ° , then

$$\langle \varphi^\circ, \psi \rangle = 0, \quad \langle \psi^\circ, \varphi \rangle = 0. \quad (25)$$

2.6 Solvability of linear operator equations

When computing the normal form coefficients in [Section 6](#) using [\(HOM\)](#), we will encounter linear operator equations of the form

$$(zI - A^{\circ*})(v_0, v) = (w_0, w), \quad (26)$$

where z is a complex number, $(w_0, w) \in X^{\circ*}$ is given and $(v_0, v) \in D(A^{\circ*})$ is the unknown. In general, both z and the right-hand side will have a nontrivial imaginary part, so here and from here onward, it is necessary to regard systems of the form [\(26\)](#) as the complexification of the original operator equations.

We will however not attach additional subscripts to the operator symbols, hoping that this omission will not cause confusion.

Since $\sigma(A)$ consists exclusively of point spectrum, there are two situations to consider depending on whether or not z is an eigenvalue. If z is *not* an eigenvalue of A then z belongs to the resolvent set $\rho(A)$ of A and (26) admits a unique solution,

$$(v_0, v) = (zI - A^{\odot*})^{-1}(w_0, w).$$

In order to actually find this solution, one needs a representation of the resolvent operator of $A^{\odot*}$. The general result can be found in [14, Corollary IV.5.4], but here we only require a special case.

Lemma 3. *Suppose that z is not an eigenvalue of A , so (26) has a unique solution (v_0, v) . If the right-hand side is represented by*

$$(w_0, w) = (w_0, \theta \mapsto e^{z\theta} \Delta^{-1}(z)\eta),$$

for some fixed vector $\eta \in \mathbb{C}^n$, then this solution has the representation

$$v_0 = v(0), \quad v(\theta) = \Delta^{-1}(z) (e^{z\theta} w_0 + (\Delta'(z) - I - \theta \Delta(z)) w(\theta)).$$

Proof. Write $(w_0, w) = (w_0, 0) + (0, \theta \mapsto e^{z\theta} \Delta^{-1}(z)\eta)$, use the linearity of $(zI - A^{\odot*})^{-1}$ and apply both cases of [37, Corollary 3.4]. \square

On the other hand, suppose that $z = \lambda$ is an eigenvalue. Then (26) need not be consistent. In fact, a solution exists if and only if

$$\langle (w_0, w), \varphi^{\odot} \rangle = 0, \quad \forall \varphi^{\odot} \in \mathcal{N}(\lambda I - A^*), \quad (\text{FSC})$$

see [37, Lemma 3.2]). This condition is often referred to as the *Fredholm solvability condition*. We note that the duality pairing in (FSC) may be evaluated in concrete cases using (22) and the complexification of (17). This will be done many times in Section 6 when we apply (FSC) to specific operator equations.

If $z = \lambda$ is an eigenvalue and (26) is consistent, then clearly its solutions are not unique. The bordered *operator* inverse

$$(\lambda I - A^{\odot*})^{\text{INV}} : \mathcal{R}(\lambda I - A^{\odot*}) \rightarrow \mathcal{D}(A^{\odot*}),$$

is used to select a particular solution in a systematic and convenient way. For the case that λ is a *simple* eigenvalue, it assigns the unique solution of the extended linear system

$$(\lambda I - A^{\odot*})(v_0, v) = (w_0, w), \quad \langle (v_0, v), \varphi^{\odot} \rangle = 0, \quad (27)$$

to every (w_0, w) for which (26) is consistent. The following lemma gives an explicit representation for a special case [37, Proposition 3.6 and Corollary 3.7].

Lemma 4. *Let $z = \lambda$ be a simple eigenvalue with eigenvector φ and adjoint eigenvector φ^{\odot} as in (22), normalized to $\langle \varphi^{\odot}, \varphi \rangle = 1$. Suppose (26) is consistent for a given right-hand side of the form*

$$(w_0, w) = (\eta, 0) + \kappa(q, \varphi),$$

where $\eta \in \mathbb{C}^n$ and $\kappa \in \mathbb{C}$. Then the unique solution (v_0, v) of (27) is given by

$$v_0 = \xi + \gamma q, \quad v(\theta) = e^{\lambda\theta} (v_0 - \kappa\theta q),$$

with $\xi = \Delta^{\text{INV}}(\lambda)(\eta + \kappa\Delta'(\lambda)q)$ and $\gamma = -p\Delta'(\lambda)\xi + \frac{1}{2}\kappa p\Delta''(\lambda)q$.

In [Section 6](#) we will use the shorthand notation

$$v = B_\lambda^{\text{INV}}(\eta, \kappa),$$

for the solution in [Lemma 4](#). We observe that the expression for ξ itself involves a bordered *matrix* inverse,

$$\Delta^{\text{INV}}(\lambda) : \mathcal{R}(\Delta(\lambda)) \rightarrow \mathbb{C}^n,$$

which assigns the unique solution of the extended linear system

$$\Delta(\lambda)x = y, \quad p \cdot x = 0,$$

to every $y \in \mathbb{C}^n$ for which the system $\Delta(\lambda)x = y$ is consistent - also see [Remark 2](#) for the notation. In practice, $x = \Delta^{\text{INV}}(\lambda)y$ can be obtained by solving the nonsingular bordered *matrix* system

$$\begin{pmatrix} \Delta(\lambda) & q \\ p & 0 \end{pmatrix} \begin{pmatrix} x \\ s \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix},$$

for the unknown $(x, s) \in \mathbb{C}^{n+1}$ that necessarily satisfies $s = 0$. The properties of (finite dimensional) bordered linear systems and their role in numerical bifurcation analysis are discussed more extensively in [\[41\]](#) and [\[28, Chapter 3\]](#).

3 Parameter dependence and classical DDEs

In [Section 3.1](#) we motivate our approach by explaining why the standard literature result does not apply to the problem at hand. This is most easily done at the concrete level of classical DDEs. The structure of the remaining subsections parallels that of [Section 2](#). Namely, we first solve the problem of parameter dependence at the more abstract level of dual perturbation theory. In the final [Section 3.6](#) we then return to classical DDEs to see how the general results apply in this special case.

3.1 Motivation

We are concerned with the situation where the right-hand side of [\(DDE\)](#) depends explicitly on parameters. Specifically, we consider

$$\dot{x}(t) = F(x_t, \alpha), \quad t \geq 0, \tag{28}$$

where $F : X \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is C^k -smooth for some $k \geq 1$ with $F(0, 0) = 0$. We assume that at the critical parameter value $\alpha = 0$ the linearization of [\(28\)](#) has $1 \leq n_0 < \infty$ purely imaginary eigenvalues, counting multiplicities. The goal of [Section 3](#) is to obtain a parameter-dependent family of local center manifolds for a class of evolution equations that includes [\(28\)](#).

In [\[14, Section IX.9.1\]](#) this problem is approached as follows. One augments [\(28\)](#) with a trivial equation for the constant parameter dynamics. This gives the system

$$\begin{cases} \dot{x}(t) = F(x_t, \mu(t)), \\ \dot{\mu}(t) = 0, \end{cases} \quad t \geq 0, \tag{29}$$

on the state space $\mathbf{X} := X \times \mathbb{R}^p$, with $X = C([-h, 0], \mathbb{R}^n)$ as before in [Section 2.4](#). Then the right-hand side of the first equation of [\(29\)](#) is split as

$$F(\varphi, \alpha) = D_1 F(0, 0)\varphi + \tilde{G}(\varphi, \alpha), \tag{30}$$

which defines $\tilde{G} : \mathbf{X} \rightarrow \mathbb{R}^n$, cf. [\[14, \(9.7\) in Section IX.9.1\]](#). The first term on the right of [\(30\)](#) acts only on the X -component of the state in \mathbf{X} , so the semigroup \tilde{T} on \mathbf{X} obtained by perturbing the shift-semigroup T_0 is *diagonal*.

However, there is an obstruction. In order to satisfy the hypotheses of the parameter-*independent* center manifold theorem, \tilde{G} must be a pure nonlinearity on \mathbf{X} , i.e.

$$\tilde{G}(0,0) = 0, \quad D_1\tilde{G}(0,0) = 0, \quad D_2\tilde{G}(0,0) = 0.$$

The first two of these conditions are clearly fulfilled, but in general there is no reason for the third condition to be met. It *does* hold when $\tilde{G}(0,\alpha) = 0$ for all $\alpha \in \mathbb{R}^p$ in a neighborhood of zero, i.e. when the zero equilibrium of (28) persists under small parameter variations. For a generic fold-Hopf bifurcation - as well as for a generic Bogdanov-Takens bifurcation that we do not discuss here - *there is no such persistence*.

In this article, the above difficulty is resolved by considering instead of (30) the splitting

$$F(\varphi, \alpha) = D_1F(0,0)\varphi + D_2F(0,0)\alpha + G(\varphi, \alpha). \quad (31)$$

Using this splitting, (29) is written as

$$\begin{cases} \dot{x}(t) = D_1F(0,0)x_t + D_2F(0,0)\mu(t) + G(x_t, \mu(t)), \\ \dot{\mu}(t) = 0, \end{cases} \quad t \geq 0. \quad (32)$$

Now *both* $D_1F(0,0)$ *as well as* $D_2F(0,0)$ appear in the perturbation of \mathbf{T}_0 . As a consequence the perturbed semigroup \mathbf{T} is no longer diagonal, but still simple enough for a complete analysis. Moreover,

$$G(0,0) = 0, \quad D_1G(0,0) = 0, \quad D_2G(0,0) = 0, \quad (33)$$

so the parameter-independent center manifold theorem can be applied without having to assume equilibrium persistence. Of course $G = \tilde{G}$ and $\mathbf{T} = \tilde{\mathbf{T}}$ whenever $D_2F(0,0) = 0$.

Remark 5. In a first attempt we regarded the augmented system (29) as a DDE on the state space $C([-h,0], \mathbb{R}^{n+p})$, also see [30], but we found this approach to be a bit unsatisfactory: The proofs in Sections 3.2 to 3.5 do not depend on the details of the class of equations under consideration, while those same details sometimes lead to notation that is more complicated than necessary.

3.2 Duality structure and linear perturbation

We work in the setting of Section 2.1. Namely, let T_0 be a \mathcal{C}_0 -semigroup on a real or complex Banach space X that is sun-reflexive with respect to T_0 . We write $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ for the underlying scalar field - as in Remark 2. Define \mathbf{T}_0 on \mathbf{X} by

$$\mathbf{T}_0(t) := \text{diag}(T_0(t), I_p). \quad (34)$$

The procedure of taking adjoints and restrictions (twice) yields semigroups \mathbf{T}_0^* , \mathbf{T}_0° , $\mathbf{T}_0^{\circ*}$ and $\mathbf{T}_0^{\circ\circ}$ on $\mathbf{X}^* \simeq X^* \times \mathbb{K}^p$, $\mathbf{X}^\circ \simeq X^\circ \times \mathbb{K}^p$, $\mathbf{X}^{\circ*} \simeq X^{\circ*} \times \mathbb{K}^p$ and $\mathbf{X}^{\circ\circ} \simeq X^{\circ\circ} \times \mathbb{K}^p$. (The symbol \simeq indicates an identification via a natural isometric isomorphism.) It is straightforward to check that on $\mathbf{X}^{\circ*}$ we have

$$\mathbf{T}_0^{\circ*}(t) = \text{diag}(T_0^{\circ*}(t), I_p), \quad (35)$$

and that the canonical injection $j : \mathbf{X} \rightarrow \mathbf{X}^{\circ*}$ has the form

$$j = \text{diag}(j, I_p), \quad (36)$$

where $j : X \rightarrow X^{\circ*}$ is the canonical injection from (2). In particular, \mathbf{X} is sun-reflexive with respect to \mathbf{T}_0 .

As in Section 2.1 we now introduce a bounded linear perturbation $L : \mathbf{X} \rightarrow \mathbf{X}^{\circ*}$ of \mathbf{T}_0 . We let it be of the form

$$L = \begin{pmatrix} L & L_p \\ 0 & 0 \end{pmatrix}, \quad (37)$$

with $L : X \rightarrow X^{\circ*}$ and $L_p : \mathbb{K}^p \rightarrow X^{\circ*}$ bounded linear operators. Perturbing T_0 by L and \mathbf{T}_0 by L yields \mathcal{C}_0 -semigroups T on X and \mathbf{T} on \mathbf{X} , respectively. Let A and \mathbf{A} be their generators.

Remark 6. There are at least two equivalent ways to compute \mathbf{T} and \mathbf{A} on \mathbf{X} and their weak* counterparts $\mathbf{T}^{\odot*}$ and $\mathbf{A}^{\odot*}$ on $\mathbf{X}^{\odot*}$. One approach - suggested to us by Odo Diekmann - uses integrated semigroup theory to calculate first \mathbf{T} and next $\mathbf{T}^{\odot*}$. Then $\mathbf{A}^{\odot*}$ and \mathbf{A} are calculated, in that order.

Here we go the other way around: We start by calculating $\mathbf{A}^{\odot*}$ and use it to obtain $\mathbf{T}^{\odot*}$. If desired \mathbf{A} and \mathbf{T} can then be found by application of (36) and its inverse. This approach is more elementary - we use only theory that was already introduced in Section 2.1 - and it yields the same outcome, as it should.

Proposition 7. *The weak* generator $\mathbf{A}^{\odot*}$ of $\mathbf{T}^{\odot*}$ has the representation*

$$\mathcal{D}(\mathbf{A}^{\odot*}) = \mathcal{D}(A^{\odot*}) \times \mathbb{K}^p, \quad \mathbf{A}^{\odot*} = \begin{pmatrix} A^{\odot*} & L_p \\ 0 & 0 \end{pmatrix},$$

where $A^{\odot*}$ is the weak* generator of $T^{\odot*}$.

Proof. According to the general theory of Section 2.1 and (4) in particular, we have

$$\mathcal{D}(\mathbf{A}^{\odot*}) = \mathcal{D}(\mathbf{A}_0^{\odot*}), \quad \mathbf{A}^{\odot*} = \mathbf{A}_0^{\odot*} + \mathbf{L}\mathbf{j}^{-1},$$

From (35) we see that

$$\mathcal{D}(\mathbf{A}_0^{\odot*}) = \mathcal{D}(A_0^{\odot*}) \times \mathbb{K}^p, \quad \mathbf{A}_0^{\odot*} = \text{diag}(A_0^{\odot*}, 0).$$

Using (36) and (37) we calculate

$$\mathbf{A}_0^{\odot*} + \mathbf{L}\mathbf{j}^{-1} = \begin{pmatrix} A_0^{\odot*} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} L & L_p \\ 0 & 0 \end{pmatrix} \begin{pmatrix} j^{-1} & 0 \\ 0 & I_p \end{pmatrix},$$

and the result follows. □

Lemma 8. *Let $L_p^{\odot} : X^{\odot} \rightarrow \mathbb{K}^p$ be the restriction of L_p^* to X^{\odot} . Then $L_p^{\odot*} = L_p$.*

Proof. We begin by noting that - strictly speaking - this statement involves two canonical identifications. Namely, let $i : X^{\odot} \rightarrow X^{\odot**}$ and $i_p : \mathbb{K}^p \rightarrow \mathbb{K}^{p**}$ be the canonical injection and bijection, respectively. Then $L_p^{\odot*} := L_p^*i$ and we need to prove that $L_p^{\odot*}i_p = L_p$. For this it is not difficult to show that

$$\langle L_p^{\odot*}i_p\alpha, \varphi^{\odot} \rangle = \langle L_p\alpha, \varphi^{\odot} \rangle,$$

for all $\alpha \in \mathbb{K}^p$ and for all $\varphi^{\odot} \in X^{\odot}$. □

For the purpose of notation, we define the *integrated semigroup* $W^{\odot*}$ for $T^{\odot*}$ as

$$W^{\odot*}(t)\varphi^{\odot*} := \int_0^t T^{\odot*}(\tau)\varphi^{\odot*} d\tau, \quad t \geq 0. \quad (38)$$

with on the right a weak* Riemann integral of the same type as the integral in (3).

Proposition 9. *The semigroup $\mathbf{T}^{\odot*}$ that is weakly* generated by $\mathbf{A}^{\odot*}$ has the representation*

$$\mathbf{T}^{\odot*}(t) = \begin{pmatrix} T^{\odot*}(t) & W^{\odot*}(t)L_p \\ 0 & I_p \end{pmatrix}, \quad t \geq 0. \quad (39)$$

Proof. We define a one-parameter family \mathbf{S} of bounded linear operators on \mathbf{X}^{\odot} by

$$\mathbf{S}(t) = \begin{pmatrix} T^{\odot}(t) & 0 \\ L_p^{\odot}W^{\odot}(t) & I_p \end{pmatrix}, \quad t \geq 0,$$

where

$$W^\circ(t)\varphi^\circ := \int_0^t T^\circ(\tau)\varphi^\circ d\tau, \quad t \geq 0.$$

It is easy to check that \mathbf{S} is a \mathcal{C}_0 -semigroup on \mathbf{X}° . By [Lemma 8](#) the adjoint semigroup $\mathbf{S}^*(t)$ equals the right-hand side of [\(39\)](#) for all $t \geq 0$. We will show that the weak* generator $\mathbf{A}^{\circ*}$ of $\mathbf{T}^{\circ*}$ is also the weak* generator of \mathbf{S}^* . This will then imply that $\mathbf{T}^{\circ*} = \mathbf{S}^*$.

We use [Proposition 7](#). Let \mathbf{C} be the generator of \mathbf{S} , so \mathbf{C}^* is the weak* generator of \mathbf{S}^* . For any $(\varphi^{\circ*}, \alpha)$ in $\mathbf{X}^{\circ*}$ and any $t > 0$ we have

$$\frac{1}{t} (\mathbf{S}^*(t)(\varphi^{\circ*}, \alpha) - (\varphi^{\circ*}, \alpha)) = \frac{1}{t} \begin{pmatrix} T^{\circ*}(t)\varphi^{\circ*} - \varphi^{\circ*} \\ 0 \end{pmatrix} + \frac{1}{t} \begin{pmatrix} W^{\circ*}(t)L_p\alpha \\ 0 \end{pmatrix}.$$

We note that $t^{-1}W^{\circ*}(t)L_p\alpha \rightarrow L_p\alpha$ weakly* as $t \downarrow 0$. It follows that the right-hand side converges weakly* if and only if $\varphi^{\circ*} \in D(A^{\circ*})$ and in that case the weak*-limit equals $(A^{\circ*}\varphi^{\circ*} + L_p\alpha, 0) = A^{\circ*}(\varphi^{\circ*}, \alpha)$. We conclude that $\mathbf{C}^* = A^{\circ*}$. \square

3.3 Spectral theory and the center eigenspace

Let T_0 be a \mathcal{C}_0 -semigroup on a complex Banach space X that is sun-reflexive with respect to T_0 . For the purpose of spectral theory, we explicitly take \mathbb{C} as the underlying scalar field. In examples, X will often be a complexification of a real Banach space, see [Remark 1](#).

We are interested in a description of the spectrum and the corresponding (generalized) eigenspaces of the generator \mathbf{A} of \mathbf{T} . In particular, [Propositions 11](#) and [14](#) below guarantee, respectively, the existence and smooth parametrization of the parameter-dependent local center manifold in [Section 3.5](#).

Proposition 10. *The spectrum $\sigma(\mathbf{A}^{\circ*}) = \sigma(A^{\circ*}) \cup \{0\}$ with resolvent operator*

$$R_z(\mathbf{A}^{\circ*}) = \begin{pmatrix} R_z(A^{\circ*}) & z^{-1}R_z(A^{\circ*})L_p \\ 0 & z^{-1}I_p \end{pmatrix}, \quad (40)$$

for every z in the resolvent set $\rho(\mathbf{A}^{\circ*})$.

Proof. From [Proposition 7](#) we have

$$zI - \mathbf{A}^{\circ*} = \begin{pmatrix} zI - A^{\circ*} & -L_p \\ 0 & zI_p \end{pmatrix}.$$

This upper triangular operator matrix has a bounded inverse if and only if *both* entries on its diagonal have bounded inverses, which happens if and only if $z \in \rho(A^{\circ*})$ and $z \neq 0$. In that case, the inverse is given precisely by the stated expression for $R_z(\mathbf{A}^{\circ*}) := (zI - \mathbf{A}^{\circ*})^{-1}$. \square

In addition we assume that T_0 is eventually compact and the perturbation L in [\(37\)](#) is compact. As a consequence, the spectral analysis of $\mathbf{A}^{\circ*}$ reduces to an analysis of the poles of its resolvent operator [[19](#), Corollary V.3.2], [[58](#), Section V.10].

Proposition 11. *\mathbf{T} is an eventually compact \mathcal{C}_0 -semigroup.*

Proof. The eventual compactness of T_0 , the finite rank of I_p and [\(34\)](#) together imply that \mathbf{T}_0 is eventually compact. Since L_p has finite rank and L is compact by assumption, it follows from [\(37\)](#) that \mathbf{L} is compact, so \mathbf{T} is eventually compact by [[11](#), Theorem 2.8]. \square

Theorem 12. *The generalized eigenspace corresponding to $\lambda \in \sigma(\mathbf{A}^{\circ*})$ is given by*

$$\mathcal{M}_\lambda(\mathbf{A}^{\circ*}) = \begin{cases} \mathcal{M}_\lambda(A^{\circ*}) \times \{0\}, & \text{if } \lambda \neq 0, \\ \mathcal{M}_0(A^{\circ*}) \times \{0\} \oplus \{(\Gamma_0 L_p \alpha, \alpha) : \alpha \in \mathbb{C}^p\}, & \text{if } \lambda = 0, \end{cases}$$

where Γ_0 is a bounded linear operator on $X^{\circ*}$ mapping into $X^{\circ\circ}$.

Proof. Let $\lambda \in \sigma(\mathbf{A}^{\odot*})$ be arbitrary. Taking residues at $z = \lambda$ in (40), we obtain

$$\mathbf{P}_\lambda^{\odot*} = \begin{pmatrix} P_\lambda^{\odot*} & \Gamma_\lambda L_p \\ 0 & I_p \delta_\lambda \end{pmatrix}, \quad \Gamma_\lambda := \operatorname{Res}_{z=\lambda} z^{-1} R_z(A^{\odot*}), \quad (41)$$

where $\delta_\lambda := \delta_{\lambda,0}$ is the Kronecker delta and $\mathbf{P}_\lambda^{\odot*}$ and $P_\lambda^{\odot*}$ are the spectral projectors corresponding to λ for $\mathbf{A}^{\odot*}$ and $A^{\odot*}$. (If λ is in the resolvent set of the respective operator, then the residue - hence the spectral projector - is identically zero.) Γ_λ is pointwise equal to a contour integral with an integrand in the closed subspace $X^{\odot\odot}$ of $X^{\odot*}$, so Γ_λ maps into $X^{\odot\odot}$.

We will now calculate the range of $\mathbf{P}_\lambda^{\odot*}$ from (41). In general,

$$\mathcal{M}_\lambda(\mathbf{A}^{\odot*}) = \left\{ \begin{pmatrix} P_\lambda^{\odot*} \varphi^{\odot*} \\ 0 \end{pmatrix} + \begin{pmatrix} \Gamma_\lambda L_p \alpha \\ \alpha \delta_\lambda \end{pmatrix} : (\varphi^{\odot*}, \alpha) \in \mathbf{X}^{\odot*} \right\}. \quad (42)$$

First we assume that $\lambda \neq 0$, so $\delta_\lambda = 0$. We are going to show that

$$\left\{ P_\lambda^{\odot*} \varphi^{\odot*} + \Gamma_\lambda L_p \alpha : (\varphi^{\odot*}, \alpha) \in \mathbf{X}^{\odot*} \right\} = \mathcal{M}_\lambda(A^{\odot*}). \quad (43)$$

Together with (42) this will then prove the theorem for $\mathcal{M}_\lambda(\mathbf{A}^{\odot*})$. To verify (43) let $p \in \mathbb{N}$ be the order of λ as a pole of $R_z(A^{\odot*})$. For $n = 1, \dots, p$ let B_n be the coefficient of $(z - \lambda)^{-n}$ in the Laurent series for $R_z(A^{\odot*})$. A small computation shows that

$$\Gamma_\lambda = \sum_{k=1}^p (-1)^{k+1} \lambda^{-k} B_k. \quad (44)$$

From [58, Section V.10] we recall the relation $B_{n+1} = (A^{\odot*} - \lambda I)^n B_1$ for all $n \in \mathbb{N}$. Since $B_1 = P_\lambda^{\odot*}$ and its range $\mathcal{M}_\lambda(A^{\odot*})$ is an invariant subspace of $A^{\odot*}$, this relation implies that B_k takes values in $\mathcal{M}_\lambda(A^{\odot*})$ for all $k = 1, \dots, p$, so the same is true for Γ_λ by (44). From this it follows that (43) holds.

For the remaining case $\lambda = 0$ we have $\delta_\lambda = 1$, so from (42) we get the direct sum

$$\mathcal{M}_0(\mathbf{A}^{\odot*}) = \left\{ \begin{pmatrix} P_0^{\odot*} \varphi^{\odot*} \\ 0 \end{pmatrix} : \varphi^{\odot*} \in X^{\odot*} \right\} \oplus \left\{ \begin{pmatrix} \Gamma_0 L_p \alpha \\ \alpha \end{pmatrix} : \alpha \in \mathbb{C}^p \right\}.$$

The first summand equals $\mathcal{M}_0(A^{\odot*}) \times \{0\}$ and this gives the result. \square

Corollary 13. *The center eigenspace \mathbf{X}_0 corresponding to the purely imaginary eigenvalues of \mathbf{A} is given by*

$$\mathbf{X}_0 = X_0 \times \{0\} \oplus \{(j^{-1} \Gamma_0 L_p \alpha, \alpha) : \alpha \in \mathbb{C}^p\},$$

with $\dim \mathbf{X}_0 = \dim X_0 + p$.

Proof. By Proposition 10 we have the disjoint union $\sigma(\mathbf{A}^{\odot*}) = (\sigma(A^{\odot*}) \setminus \{0\}) \cup \{0\}$. Using this and Theorem 12 we first compute the center eigenspace for $\mathbf{A}^{\odot*}$ as

$$\mathbf{X}_0^{\odot*} = X_0^{\odot*} \times \{0\} \oplus \{(\Gamma_0 L_p \alpha, \alpha) : \alpha \in \mathbb{C}^p\},$$

and then we apply j^{-1} from (36) to both sides of this equality. \square

In Sections 3.4 to 3.6 we will consider nonlinear problems on a real Banach space. In this case spectral analysis must be preceded by complexification, see Remark 1 and in particular [14, last part of Section IV.2].

Proposition 14. *Suppose that $X = Y_{\mathbb{C}}$ is a complexification of a real Banach space Y and let $Y_0 \subseteq Y$ be the real center eigenspace associated with X_0 . Then the real center eigenspace \mathbf{Y}_0 associated with \mathbf{X}_0 is*

$$\mathbf{Y}_0 = Y_0 \times \{0\} \oplus \{(Q\alpha, \alpha) : \alpha \in \mathbb{R}^p\} \subseteq Y \times \mathbb{R}^p, \quad (45)$$

where $Q : \mathbb{R}^p \rightarrow Y$ is a bounded linear operator. Furthermore, $\iota : \mathbf{Y}_0 \rightarrow Y_0 \times \mathbb{R}^p$ defined by $\iota(\psi, \alpha) := (\psi - Q\alpha, \alpha)$ is a linear isomorphism.

Proof. \mathbf{X} is naturally identified with $\mathbf{Y}_{\mathbb{C}}$ where $\mathbf{Y} = Y \times \mathbb{R}^p$. Let \mathbf{P}_{Λ} with range \mathbf{X}_0 be the spectral projector on \mathbf{X} for the spectral set $\Lambda = \bar{\Lambda}$ of all purely imaginary eigenvalues of $\mathbf{A}_{\mathbb{C}}$. A direct generalization of [14, Corollary IV.2.19] implies that \mathbf{P}_{Λ} is the complexification of a projector \mathbf{P}_{Λ}^Y on \mathbf{Y} and the range \mathbf{Y}_0 of \mathbf{P}_{Λ}^Y - identified with a subspace of \mathbf{Y} - is the real center eigenspace for \mathbf{A} . Also, Γ_0 on $X^{\odot\star}$ is self-conjugate by (41). Together with Corollary 13 this implies (45). It is easily verified that the linear operator ι is an isomorphism. \square

Remark 15. We will not make a *notational* distinction between the real and complex center eigenspaces, indicating both X_0 and Y_0 with X_0 and both \mathbf{X}_0 and \mathbf{Y}_0 with \mathbf{X}_0 , respectively. We hope that the underlying scalar field will be clear from the immediate context.

3.4 Nonlinear perturbation

Let T_0 be a \mathcal{C}_0 -semigroup on a *real* Banach space X that is sun-reflexive with respect to T_0 . Introduce a nonlinear perturbation $\mathbf{R} : \mathbf{X} \rightarrow \mathbf{X}^{\odot\star}$ of the form

$$\mathbf{R}(\varphi, \alpha) = (R(\varphi, \alpha), 0), \quad (46)$$

where $R : \mathbf{X} \rightarrow X^{\odot\star}$ is C^k -smooth, satisfying

$$R(0, 0) = 0, \quad D_1 R(0, 0) = 0, \quad D_2 R(0, 0) = 0. \quad (47)$$

We associate with \mathbf{T} and \mathbf{R} the integral equation

$$\mathbf{u}(t) = \mathbf{T}(t-s)\mathbf{u}(s) + \mathbf{j}^{-1} \int_s^t \mathbf{T}^{\odot\star}(t-\tau)\mathbf{R}(\mathbf{u}(\tau)) d\tau, \quad -\infty < s \leq t < \infty. \quad (48)$$

We expect all nontrivial dynamics to be contained in the first component, and this is indeed the case:

Proposition 16. *The function $\mathbf{u} = (u, u_p) : I \rightarrow \mathbf{X}$ is a solution of (48) if and only if u_p is constant on I and $u : I \rightarrow X$ is a solution of*

$$u(t) = T(t-s)u(s) + \mathbf{j}^{-1} \int_s^t T^{\odot\star}(t-\tau)(L_p \alpha + R(u(\tau), \alpha)) d\tau, \quad -\infty < s \leq t < \infty, \quad (49)$$

where $\alpha \in \mathbb{R}^p$ denotes the constant value of u_p .

Proof. We use Proposition 9 and (36). For any continuous function $\mathbf{u} = (u, u_p) : I \rightarrow \mathbf{X}$ we compute

$$\mathbf{T}(t-s)\mathbf{u}(s) = \mathbf{j}^{-1} \mathbf{T}^{\odot\star}(t-s) \mathbf{j} \mathbf{u}(s) = \begin{pmatrix} T(t-s)u(s) + \mathbf{j}^{-1} W^{\odot\star}(t-s) L_p u_p(s) \\ u_p(s) \end{pmatrix},$$

while another computation shows that

$$\mathbf{j}^{-1} \int_s^t \mathbf{T}^{\odot\star}(t-\tau) \mathbf{R}(\mathbf{u}(\tau)) d\tau = \begin{pmatrix} \mathbf{j}^{-1} \int_s^t T^{\odot\star}(t-\tau) R(\mathbf{u}(\tau)) d\tau \\ 0 \end{pmatrix}.$$

From the above together with the definition (38) of $W^{\odot\star}(t)$ we see that (48) is equivalent to the system

$$\begin{cases} u(t) = T(t-s)u(s) + \mathbf{j}^{-1} \int_s^t T^{\odot\star}(t-\tau)(L_p u_p(s) + R(u(\tau), u_p(\tau))) d\tau, \\ u_p(t) = u_p(s), \end{cases}$$

for $-\infty < s \leq t < \infty$. The statement now follows. \square

3.5 Parameter-dependent local center manifolds

We consider again a \mathcal{C}_0 -semigroup T_0 on a real Banach space X that is sun-reflexive with respect to T_0 . We also assume that T_0 is eventually compact and L in (37) is compact, so Proposition 11 implies that \mathbf{T} is eventually compact. If furthermore the nonlinearity \mathbf{R} satisfies (47), then all conditions are fulfilled for the application of the center manifold theory from [14, Chapter IX] to (48).

Therefore, if the generator A of T has $1 \leq n_0 < \infty$ purely imaginary eigenvalues, counting algebraic multiplicities, then by Proposition 14 the real center eigenspace \mathbf{X}_0 has dimension $n_0 + p$. There exists a C^k -smooth local center manifold $\mathcal{W}_{\text{loc}}^c$ in \mathbf{X} that is tangent at the origin to \mathbf{X}_0 . In fact, Proposition 14 implies that $\mathcal{W}_{\text{loc}}^c$ is the image of a C^k -smooth map

$$\mathcal{C} : U \times U_p \subseteq X_0 \times \mathbb{R}^p \rightarrow \mathbf{X},$$

where $U \subseteq X_0$ and $U_p \subseteq \mathbb{R}^p$ are neighborhoods of the origin. Since (46) has a zero in the second component, it follows from [14, (5.1) in Section IX.5] that \mathcal{C} has the form

$$\mathcal{C}(\varphi, \alpha) = (\mathcal{C}(\varphi, \alpha), \alpha), \quad \forall (\varphi, \alpha) \in U \times U_p, \quad (50)$$

where $\mathcal{C} : U \times U_p \rightarrow X$ is the first component function.

Definition 17. The image $\mathcal{W}_{\text{loc}}^c(\alpha) := \mathcal{C}(U, \alpha)$ is a *local center manifold* for (49) at $\alpha \in U_p$.

It is a direct consequence of the above definition that for every $\alpha \in U_p$ we can parametrize $\mathcal{W}_{\text{loc}}^c(\alpha)$ by coordinates on the real center eigenspace X_0 that depend C^k -smoothly on α . This will be important for the discussion of the normalization method following (55) in Section 4.

Proposition 18. *If $\alpha \in U_p$ is sufficiently small then $\mathcal{W}_{\text{loc}}^c(\alpha)$ is locally positively invariant for the semiflow generated by (49).*

Proof. Let Σ and Σ be the semiflows generated by (48) and (49), respectively. By Proposition 16,

$$\Sigma(s, (\psi, \alpha)) = (\Sigma(s, \psi), \alpha), \quad \forall \psi \in X, \alpha \in \mathbb{R}^p, \quad (51)$$

and for all s in a common interval of existence $I_{\psi, \alpha}$. By [14, Theorem IX.5.3(i)] there exists $\delta > 0$ such that if $(\psi, \alpha) \in \mathcal{W}_{\text{loc}}^c$ and if

$$\|\Sigma(s, (\psi, \alpha))\| = \|\Sigma(s, \psi)\| + |\alpha| \leq \delta, \quad \forall s \in [0, t],$$

then $\Sigma(t, (\psi, \alpha)) \in \mathcal{W}_{\text{loc}}^c$ which by (51) implies that $\Sigma(t, \psi) \in \mathcal{W}_{\text{loc}}^c(\alpha)$.

We note that if $\psi \in \mathcal{W}_{\text{loc}}^c(\alpha)$ then by (50) it follows that $(\psi, \alpha) \in \mathcal{W}_{\text{loc}}^c$. Therefore, if $|\alpha| \leq \frac{\delta}{2}$ and $\psi \in \mathcal{W}_{\text{loc}}^c(\alpha)$ then

$$\|\Sigma(s, \psi)\| \leq \frac{\delta}{2}, \quad \forall s \in [0, t]$$

implies that $\Sigma(t, \psi) \in \mathcal{W}_{\text{loc}}^c(\alpha)$. This is precisely local positive invariance of $\mathcal{W}_{\text{loc}}^c(\alpha)$ for Σ . \square

Next we consider a solution $u : I \rightarrow X$ of (49) that lies in $\mathcal{W}_{\text{loc}}^c(\alpha)$. By Proposition 16 the function $\mathbf{u} = (u, \alpha) : I \rightarrow \mathbf{X}$ is a solution of (48). Also, since u lies in $\mathcal{W}_{\text{loc}}^c(\alpha)$ we see from (50) that \mathbf{u} lies in $\mathcal{W}_{\text{loc}}^c$ and therefore satisfies the differential equation

$$j\dot{\mathbf{u}}(t) = A^{\odot*} j\mathbf{u}(t) + \mathbf{R}(\mathbf{u}(t)), \quad \forall t \in I,$$

cf. (9). By (36) and (46) and Proposition 7 the first component of this equation gives the differential equation

$$j\dot{u}(t) = A^{\odot*} ju(t) + L_p \alpha + R(u(t), \alpha), \quad \forall t \in I, \quad (52)$$

that is satisfied by u .

In summary,

Theorem 19 (Parameter-dependent local center manifold). *Let T_0 be an eventually compact \mathcal{C}_0 -semigroup on a sun-reflexive real Banach space X and let T be the \mathcal{C}_0 -semigroup on X defined by (3) where L is a compact perturbation. Suppose that the generator A of T has $1 \leq n_0 < \infty$ purely imaginary eigenvalues with corresponding n_0 -dimensional real center eigenspace X_0 . Furthermore, assume that R in (49) is C^k -smooth and (47) holds.*

Then there exists a C^k -smooth map $\mathcal{C} : U \times U_p \rightarrow X$ defined in a neighborhood of the origin in $X_0 \times \mathbb{R}^p$ and such that for every sufficiently small $\alpha \in \mathbb{R}^p$ the manifold $\mathcal{W}_{loc}^c(\alpha) := \mathcal{C}(U, \alpha)$ is locally positively invariant for the semiflow generated by (49) at parameter value α . Furthermore, any solution $u : I \rightarrow X$ of (49) that lies on $\mathcal{W}_{loc}^c(\alpha)$ satisfies (52).

3.6 The special case of parameter-dependent classical DDEs

In this section we will formulate a corollary of Theorem 19 that applies specifically to the classical parameter-dependent DDE (28). As in Section 2.4 our starting point is (32) with $F = 0$,

$$\begin{cases} \dot{x}(t) = 0, \\ \dot{\mu}(t) = 0, \end{cases} \quad t \geq 0, \quad (53)$$

in the unknown (x, μ) with initial condition (φ, α) in the state space $\mathbf{X} := X \times \mathbb{R}^p$ where $X := C([-h, 0], \mathbb{R}^n)$. So, we interpret the first component of (53) as a DDE but the second component as an ODE. By comparison with (13) it is clear that the solution of the initial value problem for (53) defines a \mathcal{C}_0 -semigroup \mathbf{T}_0 on \mathbf{X} ,

$$\mathbf{T}_0(t) := \text{diag}(T_0(t), I_p),$$

with T_0 the eventually compact shift semigroup on X from (14) and I_p the identity on \mathbb{R}^p . We specify the perturbations L and L_p in (37) as

$$L\varphi = (D_1F(0, 0)\varphi)r^{\odot*}, \quad L_p\alpha = (D_2F(0, 0)\alpha)r^{\odot*},$$

where $D_1F(0, 0)$ and $D_2F(0, 0)$ are as in (31). Then L is of finite rank, so certainly it is compact. Also, we choose the nonlinear perturbation R in (46) as

$$R(\varphi, \alpha) = G(\varphi, \alpha)r^{\odot*}, \quad (54)$$

where G is defined by the splitting in (31). Then (33) implies that the conditions in (47) hold.

Corollary 20 (Parameter-dependent local center manifold for DDEs). *Consider the particular case of the classical DDE in (28),*

$$\dot{x}(t) = F(x_t, \alpha), \quad t \geq 0,$$

where $F : X \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is C^k -smooth for some $k \geq 1$ with $F(0, 0) = 0$. Let T be the \mathcal{C}_0 -semigroup on X corresponding to the linearization of (28) at $0 \in X$ for the critical parameter value $\alpha = 0$. Suppose that the generator A of T has $1 \leq n_0 < \infty$ purely imaginary eigenvalues with corresponding n_0 -dimensional real center eigenspace X_0 .

Then there exists a C^k -smooth map $\mathcal{C} : U \times U_p \rightarrow X$ defined in a neighborhood of the origin in $X_0 \times \mathbb{R}^p$ and such that for every sufficiently small $\alpha \in \mathbb{R}^p$ the manifold $\mathcal{W}_{loc}^c(\alpha) := \mathcal{C}(U, \alpha)$ is locally positively invariant for the semiflow generated by (28) at parameter value α .

Furthermore, if the history x_t associated with a solution of (28) exists on some nondegenerate interval I and $x_t \in \mathcal{W}_{loc}^c(\alpha)$ for all $t \in I$, then $u : I \rightarrow X$ defined by $u(t) := x_t$ is differentiable and satisfies

$$j\dot{u}(t) = A^{\odot*}ju(t) + (D_2F(0, 0)\alpha)r^{\odot*} + G(u(t), \alpha)r^{\odot*}, \quad \forall t \in I,$$

where A^{\odot} is the weak* generator of $T^{\odot*}$ and the operator $G : X \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ defined by (31),*

$$G(\varphi, \alpha) := F(\varphi, \alpha) - D_1F(0, 0)\varphi - D_2F(0, 0)\alpha,$$

is the nonlinear part of F .

Remark 21. For discrete classical DDEs (1) one may want to use one or more of the discrete delays τ_1, \dots, τ_m as parameters. However, in this case the operator R is typically no longer differentiable, see [14, Remark IX.9.2]. Of course, if only a single discrete delay is present, this problem can be avoided by a linear rescaling of time.

4 Normal forms on the parameter-dependent center manifold

The normalization technique described in this section goes back to [8]. In [42] it was applied to obtain expressions for the critical normal form coefficients of all generic codimension one and two bifurcations of equilibria in ODEs, also see [43, §8.7]. In this context, these expressions are independent of the (finite) dimension of the phase space and they involve only critical eigenvectors of the Jacobian matrix and its transpose as well as higher order derivatives of the right-hand side at the critical equilibrium. These properties make them suitable for both symbolic and numerical evaluation.

In [46] the same technique was applied to parameter-dependent normal forms to start the continuation of nonhyperbolic cycles emanating from generalized Hopf, fold-Hopf and Hopf-Hopf bifurcation points of ODEs. The resulting predictors were implemented in the freely available software package `MatCont`, a MATLAB toolbox for continuation and bifurcation analysis of finite dimensional dynamical systems. This makes it possible to verify transversality conditions and to initialize the continuation of the nonhyperbolic cycles mentioned above. A similar switching problem for iterated maps was solved earlier in [27].

In [37] the normalization technique was lifted to an infinite dimensional setting, providing explicit expressions for the critical normal form coefficients of generic codimension one and two equilibrium bifurcations in DDEs. These expressions were partially implemented [60] in the software `DDE-BifTool`. This package can be considered as the DDE equivalent of `MatCont` in command line mode.

In this section we extend the normalization method from [37] to include parameters. Suppose $0 \in X$ is a stationary state of (28) at the critical parameter value $0 \in \mathbb{R}^p$ and assume there are $n_0 \geq 1$ eigenvalues on the imaginary axis, counting algebraic multiplicities. Let P_0 be the corresponding real spectral projector on X , so the range X_0 of P_0 is the real n_0 -dimensional center eigenspace. **Corollary 20** applies to give a parameter-dependent local center manifold $\mathcal{W}_{\text{loc}}^c(\alpha)$ for (28).

We allow for the introduction of a new parameter β defined in a neighborhood of $0 \in \mathbb{R}^p$ such that $\alpha = K(\beta)$ for some locally defined C^k -diffeomorphism $K : \mathbb{R}^p \rightarrow \mathbb{R}^p$ that is to be determined below, up to a certain order. If $u : I \rightarrow X$ with $u(t) := x_t \in \mathcal{W}_{\text{loc}}^c(\alpha)$ is as in **Corollary 20**, then u is differentiable on I and satisfies

$$j\dot{u}(t) = A^{\odot*}ju(t) + (D_2F(0,0)K(\beta))r^{\odot*} + R(u(t), K(\beta)), \quad \forall t \in I, \quad (55)$$

where R encodes the nonlinear part of F as in (54). Choose a basis Φ of X_0 and let $\mathcal{H} : \mathbb{R}^{n_0} \times \mathbb{R}^p \rightarrow X$ be a locally defined C^k -smooth parametrization of $\mathcal{W}_{\text{loc}}^c(\alpha)$ with respect to Φ and in terms of the new parameter β , see the remark following **Definition 17**. For every $t \in I$ we define $v(t) \in \mathbb{R}^{n_0}$ as the coordinate vector of $P_0u(t)$ with respect to Φ . Then $v : I \rightarrow \mathbb{R}^{n_0}$ satisfies a parameter-dependent ordinary differential equation of the form

$$\dot{v} = \sum_{|\nu|+|\mu| \geq 1} \frac{1}{\nu! \mu!} g_{\nu\mu} v^\nu \beta^\mu, \quad (56)$$

where the C^k -smooth vector field on the right has been expanded up to some sufficiently large - but finite - order. The multi-indices ν and μ have lengths n_0 and p , respectively. We assume that (56) is in parameter-dependent normal form, up to a certain order. Since \mathcal{H} parametrizes $\mathcal{W}_{\text{loc}}^c(\alpha)$,

$$u(t) = \mathcal{H}(v(t), \beta), \quad t \in I,$$

with both u and v depending on the parameter, although this is left implicit in the notation. Substituting the above relation into (55) produces the *homological equation*

$$A^{\odot\star} j\mathcal{H}(v, \beta) + (D_2 F(0, 0)K(\beta))r^{\odot\star} + R(\mathcal{H}(v, \beta), K(\beta)) = jD_1 \mathcal{H}(v, \beta)\dot{v}, \quad (\text{HOM})$$

with \dot{v} given by the parameter-dependent normal form (56). The unknowns in (HOM) are \mathcal{H} , K and the coefficients $g_{\nu\mu}$ from (56). They are determined, up to a certain order, by the assumption that (56) is in normal form. For $r, s \geq 0$ with $r + s \geq 1$ we denote by $D_1^r D_2^s F(0, 0) : X^r \times [\mathbb{R}^p]^s \rightarrow \mathbb{R}^n$ the mixed Fréchet derivative of order $r + s$, evaluated at $(0, 0) \in X \times \mathbb{R}^p$, with the understanding that at most one of the factor spaces X^r or $[\mathbb{R}^p]^s$ is absent if either $r = 0$ or $s = 0$. We expand the nonlinearity R as

$$R(\varphi, \alpha) = \sum_{r+s>1} \frac{1}{r!s!} D_1^r D_2^s F(0, 0)(\varphi^{(r)}, \alpha^{(s)})r^{\odot\star}, \quad (57)$$

where $\varphi^{(r)} := (\varphi, \dots, \varphi) \in X^r$ and $\alpha^{(s)} := (\alpha, \dots, \alpha) \in [\mathbb{R}^p]^s$. The mappings \mathcal{H} and K can be expanded as

$$\mathcal{H}(v, \beta) = \sum_{|\nu|+|\mu|\geq 1} \frac{1}{\nu!\mu!} H_{\nu\mu} v^\nu \beta^\mu, \quad K(\beta) = \sum_{|\mu|\geq 1} \frac{1}{\mu!} K_\mu \beta^\mu. \quad (58)$$

Substituting (56)–(58) into (HOM), collecting coefficients of terms $v^\nu \beta^\mu$ from lower to higher order and solving the resulting linear systems, one can solve recursively for the unknown coefficients $g_{\nu\mu}$, $H_{\nu\mu}$ and K_μ by applying the Fredholm alternative and taking ordinary or bordered inverses, as explained in Section 2.6.

5 Predictors for normal forms on center manifolds

Starting from this point, we will focus exclusively on *two-parameter* DDEs, i.e. we will have $p = 2$. This enables the initialization of codimension one equilibrium and cycle bifurcations emanating from the generalized Hopf, fold-Hopf, Hopf-Hopf and transcritical-Hopf codimension two bifurcation points. The corresponding steady state and cycle bifurcation curves can then be continued in two parameters using the continuation capabilities of `DDE-BifTool`, providing an unfolding of the singularity.

In this section we list known asymptotics for codimension one nonhyperbolic cycles emanating from generalized Hopf, fold-Hopf and Hopf-Hopf bifurcations obtained in [46]. The asymptotics are derived from the Poincaré normal forms which are obtained by considering near-identity changes of coordinates generated by homogeneous polynomial functions without using time reparametrization. These normal forms are therefore ready to be used in conjunction with the homological equation (HOM) where all the time derivatives are assumed to be with the same unit of time. Following the same method as in [46], we also derive asymptotics for codimension one nonhyperbolic cycles emanating from the transcritical-Hopf bifurcation. We also provide asymptotics for the codimension one equilibrium bifurcations emanating from the degenerate Hopf points.

5.1 Generalized-Hopf bifurcation

Suppose that system (28) has an equilibrium $\varphi = 0$ at the critical parameter value $\alpha_0 = (0, 0) \in \mathbb{R}^2$ with purely imaginary eigenvalues

$$\lambda_{1,2} = \pm i\omega_0, \quad \omega_0 > 0. \quad (59)$$

Furthermore, suppose that the first Lyapunov coefficient $\ell_1(0) = 0$ while the second Lyapunov coefficient $\ell_2(0) \neq 0$. Then the restriction of (28) to the two-dimensional center manifold $\mathcal{W}_{\text{loc}}^c(\alpha)$ can be transformed into the smooth local normal form

$$\dot{z} = \lambda(\alpha)z + c_1(\alpha)z|z|^2 + c_2(\alpha)z|z|^4 + \mathcal{O}(|z|^6), \quad (60)$$

where $\lambda(\alpha), c_1(\alpha), c_2(\alpha)$ are complex-valued functions with $\ell_1(0) = \frac{1}{\omega_0} \operatorname{Re} c_1(0) = 0$, $\lambda(0) = i\omega_0$ and $\ell_2(0) = \frac{1}{\omega_0} \operatorname{Re} c_2(0) \neq 0$. Let

$$\begin{cases} \lambda(\alpha) = \mu(\alpha) + i\omega(\alpha), \\ c_1(\alpha) = \operatorname{Re} c_1(\alpha) + i \operatorname{Im} c_1(\alpha), \end{cases}$$

where $\mu(\alpha)$ and $\omega(\alpha)$ are real-valued functions. Suppose that the map $\alpha \mapsto (\mu(\alpha), \operatorname{Re} c_1(\alpha))$ is regular at $\alpha = 0$. Then we can introduce new parameters $(\beta_1(\alpha), \beta_2(\alpha))$ to obtain the normal form

$$\dot{z} = (\beta_1 + i\omega(\beta))z + (\beta_2 + i \operatorname{Im} c_1(\beta))z|z|^2 + c_2(\beta)z|z|^4 + \mathcal{O}(|z|^6), \quad (61)$$

where $\omega(0) = \omega_0$. For convenience, we abuse notations and write $\omega(\beta)$ and $c_j(\beta)$ instead of $\omega(\alpha(\beta))$ and $c_j(\alpha(\beta))$, respectively. Similar conventions are adopted in other cases ahead.

It is well known that a curve of fold bifurcation of limit cycles (LPC) emanates from this codimension two point. To approximate this curve we substitute $z = \rho e^{i\psi}$ into (61) and expand $\omega(\beta) = \omega_0 + \omega_{10}\beta_1 + \omega_{01}\beta_2 + \mathcal{O}(\|(\beta_1, \beta_2)\|^2)$. Truncating higher order terms and separating the real and imaginary parts yields the system

$$\begin{cases} \dot{\rho} = \rho (\beta_1 + \beta_2 \rho^2 + \operatorname{Re}(c_2(0))\rho^4), \\ \dot{\psi} = \omega_0 + \omega_{10}\beta_1 + \omega_{01}\beta_2 + \operatorname{Im}(c_1(0))\rho^2 + \operatorname{Im}(c_2(0))\rho^4. \end{cases} \quad (62)$$

The curve LPC corresponds to a double zero in the amplitude equation in (62). Therefore, this curve in (61) can be approximated by

$$\rho = \epsilon, \quad \beta_1 = \operatorname{Re}(c_2(0))\epsilon^4, \quad \beta_2 = -2 \operatorname{Re}(c_2(0))\epsilon^2, \quad \epsilon > 0. \quad (63)$$

From the second equation in the amplitude system (62) we obtain using equation (63) the following approximation for the period:

$$T = 2\pi / (\omega_0 + (\operatorname{Im}(c_1(0)) - 2 \operatorname{Re} c_2(0)\omega_{01}) \epsilon^2). \quad (64)$$

Since $\operatorname{Re}(c_1(\beta)) = \beta_2$, it is easy to see that the Hopf curve in the original system is related to the truncated normal form by

$$(\beta_1, \beta_2, z) = (0, \epsilon, 0) \quad (65)$$

for $\epsilon \neq 0$ small.

5.2 Fold-Hopf bifurcation

Suppose that system (28) has an equilibrium $x = 0$ at the critical parameter value $\alpha_0 = (0, 0) \in \mathbb{R}^2$ with eigenvalues

$$\lambda_1 = 0, \quad \lambda_{2,3} = \pm i\omega_0, \quad (66)$$

where $\omega_0 > 0$. The restriction of (28) to the three-dimensional center manifold $\mathcal{W}_{\text{loc}}^c(\alpha)$ can generically be transformed to the smooth local normal form

$$\begin{cases} \dot{z}_0 = \gamma(\alpha) + g_{200}(\alpha)z_0^2 + g_{011}(\alpha)|z_1|^2 + g_{300}(\alpha)z_0^3 + g_{111}(\alpha)z_0|z_1|^2 \\ \quad + \mathcal{O}(\|(z_0, z_1, \bar{z}_1)\|^4), \\ \dot{z}_1 = \lambda(\alpha)z_1 + g_{110}(\alpha)z_0z_1 + g_{210}(\alpha)z_0^2z_1 + g_{021}(\alpha)z_1|z_1|^2 + \mathcal{O}(\|(z_0, z_1, \bar{z}_1)\|^4), \end{cases} \quad (67)$$

where $z_0 \in \mathbb{R}$, $z_1 \in \mathbb{C}$, $\gamma(0) = 0$, $\lambda(0) = i\omega_0$ and the functions $g_{jkl}(\alpha)$ are real in the first equation and complex in the second. Let $\lambda(\alpha) = \mu(\alpha) + i\omega(\alpha)$ and suppose that the map $\alpha \mapsto (\gamma(\alpha), \mu(\alpha))$ is regular at $\alpha = 0$. Introducing new parameters $(\beta_1(\alpha), \beta_2(\alpha))$, we obtain the truncated normal form

$$\begin{cases} \dot{z}_0 = \beta_1 + g_{200}(\beta)z_0^2 + g_{011}(\beta)|z_1|^2 + g_{111}(\beta)z_0|z_1|^2 + g_{300}(\beta)z_0^3, \\ \dot{z}_1 = (\beta_2 + i\omega_0 + ib_1(\beta))z_1 + g_{110}(\beta)z_0z_1 + g_{210}(\beta)z_0^2z_1 + g_{021}(\beta)z_1|z_1|^2, \end{cases} \quad (68)$$

with $b_1(0) = 0$. Letting $z_1 = \rho e^{i\psi}$ and separating the real and imaginary parts yields the system

$$\begin{cases} \dot{z}_0 = \beta_1 + g_{200}(\beta)z_0^2 + g_{011}(\beta)\rho^2 + g_{111}(\beta)z_0\rho^2 + g_{300}(\beta)z_0^3, \\ \dot{\rho} = \rho(\beta_2 + \operatorname{Re}(g_{110}(\beta))z_0 + \operatorname{Re}(g_{210}(\beta))z_0^2 + \operatorname{Re}(g_{021}(\beta))\rho^2), \\ \dot{\psi} = \omega_0 + b_1(\beta) + \operatorname{Im}(g_{110}(\beta))z_0 + \operatorname{Im}(g_{210}(\beta))z_0^2 + \operatorname{Im}(g_{021}(\beta))\rho^2. \end{cases} \quad (69)$$

5.2.1 Neimark-Sacker curve

A Hopf bifurcation in the amplitude system (69), i.e. when the trace of the Jacobian matrix of the amplitude system vanishes but the determinant of this matrix is positive, corresponds to a Neimark-Sacker bifurcation in the original system. Let $\rho = \epsilon$ and suppose that $g_{011}(0)\operatorname{Re}(g_{110}(0)) < 0$. Then by the implicit function theorem, we obtain the second-order predictor

$$\begin{cases} \beta_1 = -g_{011}(0)\epsilon^2, \\ \beta_2 = \frac{\operatorname{Re}(g_{110}(0))(2\operatorname{Re}(g_{021}(0)) + g_{111}(0)) - 2\operatorname{Re}(g_{021}(0))g_{200}(0)}{2g_{200}(0)}\epsilon^2, \\ z_0 = -\frac{2\operatorname{Re}(g_{021}(0)) + g_{111}(0)}{2g_{200}(0)}\epsilon^2, \end{cases} \quad (70)$$

for the Neimark-Sacker curve. It follows from (69) that the period of the cycle is approximated to the same order by

$$T = 2\pi / (\omega_0 + \omega_1\beta_1 + \omega_2\beta_2 + \operatorname{Im}(g_{110}(0))z_0 + \operatorname{Im}(g_{021}(0))\epsilon^2).$$

Here (z_0, β_1, β_2) are as in (70) and $\omega_i = \frac{\partial b_1}{\partial \beta_i}(0)$, for $i = 1, 2$.

Remark 22. Notice that the cubic terms $g_{210}(0)$ and $g_{300}(0)$ do not appear in the predictor. However, these coefficients are needed to determine the stability of the two-dimensional torus, see [43, Section 8.5] and (106).

5.2.2 Fold and Hopf curves

For the approximation to the fold and Hopf curves it is sufficient to consider the second order terms in (z_0, ρ) in the amplitude system in (69). We obtain three equilibrium solutions given by

$$\pm \left(\sqrt{-\frac{\beta_1}{g_{200}(0)}}, 0 \right) \quad \text{and} \quad \left(-\frac{\beta_2}{\operatorname{Re}(g_{110}(0))}, \frac{\sqrt{-\operatorname{Re}(g_{110}(0))^2\beta_1 - g_{200}(0)\beta_2^2}}{\sqrt{g_{011}(0)\operatorname{Re}(g_{110}(0))}} \right).$$

It follows that the fold curve is approximated by

$$(\beta_1, \beta_2) = (0, \epsilon)$$

and the Hopf curve by

$$(\beta_1, \beta_2) = \left(-\frac{g_{200}(0)}{\operatorname{Re}(g_{110}(0))^2}\epsilon^2, \epsilon \right),$$

for $|\epsilon|$ small.

5.3 Hopf-Hopf bifurcation

Suppose that system (28) at the critical parameter value $\alpha_0 = (0, 0) \in \mathbb{R}^2$ undergoes two Hopf bifurcations simultaneously. Then the generator A has two pairs of purely imaginary eigenvalues

$$\lambda_{1,4} = \pm i\omega_1(0), \quad \lambda_{2,3} = \pm i\omega_2(0), \quad (71)$$

where $\omega_i : \mathbb{R}^2 \rightarrow \mathbb{C}$ for $i = 1, 2$ such that $\omega_1(0) > \omega_2(0) > 0$. When no other eigenvalues on the imaginary axis exist, this phenomenon is called the Hopf-Hopf or double-Hopf bifurcation. Assume, furthermore that the nonresonance conditions $k\omega_1 \neq l\omega_2$, with $0 < k + l \leq 5$ are satisfied. Then the restriction of (28) to the four-dimensional center manifold $\mathcal{W}_{\text{loc}}^c(\alpha)$ can be transformed to the smooth local normal form

$$\begin{cases} \dot{z}_1 = \lambda_1(\alpha)z_1 + g_{2100}(\alpha)z_1|z_1|^2 + g_{1011}(\alpha)z_1|z_2|^2 + g_{3200}(\alpha)z_1|z_1|^4 \\ \quad + g_{2111}(\alpha)z_1|z_1|^2|z_2|^2 + g_{1022}(\alpha)z_1|z_2|^4 + \mathcal{O}(\|z_1, \bar{z}_1, z_2, \bar{z}_2\|^6), \\ \dot{z}_2 = \lambda_2(\alpha)z_2 + g_{1110}(\alpha)z_2|z_1|^2 + g_{0021}(\alpha)z_2|z_2|^2 + g_{2210}(\alpha)z_2|z_1|^4 \\ \quad + g_{1121}(\alpha)z_2|z_1|^2|z_2|^2 + g_{0032}(\alpha)z_2|z_2|^4 + \mathcal{O}(\|z_1, \bar{z}_1, z_2, \bar{z}_2\|^6), \end{cases}$$

where $z_1, z_2 \in \mathbb{C}^2$ and $g_{jklm} \in \mathbb{C}$. Let

$$\begin{cases} \lambda_1(\alpha) = \mu_1(\alpha) + i\omega_1(\alpha), \\ \lambda_2(\alpha) = \mu_2(\alpha) + i\omega_2(\alpha), \end{cases}$$

and suppose that the map $\alpha \mapsto (\mu_1(\alpha), \mu_2(\alpha))$ is regular at $\alpha = 0$. Then we can introduce new parameters $(\beta_1(\alpha), \beta_2(\alpha)) = (\mu_1(\alpha), \mu_2(\alpha))$ to obtain the normal form

$$\begin{cases} \dot{z}_1 = (\beta_1 + i\omega_1(\beta))z_1 + g_{2100}(\beta)z_1|z_1|^2 + g_{1011}(\beta)z_1|z_2|^2 + g_{3200}(\beta)z_1|z_1|^4 \\ \quad + g_{2111}(\beta)z_1|z_1|^2|z_2|^2 + g_{1022}(\beta)z_1|z_2|^4 + \mathcal{O}(\|z_1, \bar{z}_1, z_2, \bar{z}_2\|^6), \\ \dot{z}_2 = (\beta_2 + i\omega_2(\beta))z_2 + g_{1110}(\beta)z_2|z_1|^2 + g_{0021}(\beta)z_2|z_2|^2 + g_{2210}(\beta)z_2|z_1|^4 \\ \quad + g_{1121}(\beta)z_2|z_1|^2|z_2|^2 + g_{0032}(\beta)z_2|z_2|^4 + \mathcal{O}(\|z_1, \bar{z}_1, z_2, \bar{z}_2\|^6). \end{cases}$$

We truncate the normal form to third order

$$\begin{cases} \dot{z}_1 = (\beta_1 + i\omega_1 + ib_1(\beta))z_1 + g_{2100}(0)z_1|z_1|^2 + g_{1011}(0)z_1|z_2|^2, \\ \dot{z}_2 = (\beta_2 + i\omega_2 + ib_2(\beta))z_2 + g_{1110}(0)z_2|z_1|^2 + g_{0021}(0)z_2|z_2|^2. \end{cases} \quad (72)$$

Letting $(z_1, z_2) = (\rho_1 e^{i\psi_1}, \rho_2 e^{i\psi_2})$ and separating the real and imaginary parts yields

$$\begin{cases} \dot{\rho}_1 = \rho_1 (\beta_1 + \text{Re}(g_{2100}(0))\rho_1^2 + \text{Re}(g_{1011}(0))\rho_2^2), \\ \dot{\rho}_2 = \rho_2 (\beta_2 + \text{Re}(g_{1110}(0))\rho_1^2 + \text{Re}(g_{0021}(0))\rho_2^2), \\ \dot{\psi}_1 = \omega_1 + b_1(\beta) + \text{Im}(g_{2100}(0))\rho_1^2 + \text{Im}(g_{1011}(0))\rho_2^2, \\ \dot{\psi}_2 = \omega_2 + b_2(\beta) + \text{Im}(g_{1110}(0))\rho_1^2 + \text{Im}(g_{0021}(0))\rho_2^2. \end{cases} \quad (73)$$

5.3.1 Hopf curves

There are two semi-trivial equilibria

$$(\rho_1, \rho_2) = \left(\sqrt{-\frac{\beta_1}{\text{Re}(g_{2100}(0))}}, 0 \right), \quad (\rho_1, \rho_2) = \left(0, \sqrt{-\frac{\beta_2}{\text{Re}(g_{0021}(0))}} \right)$$

of the amplitude system of (73). Translating to the original system provides the Hopf bifurcation curves

$$H_1 = \{(\beta_1, \beta_2) : \beta_1 = 0\}, \quad \text{and} \quad H_2 = \{(\beta_1, \beta_2) : \beta_2 = 0\}.$$

5.3.2 Neimark-Sacker curves

A nontrivial equilibrium to the amplitude system

$$(\rho_1, \rho_2) = \left(\sqrt{\frac{\beta_2 \operatorname{Re} g_{1011}(0) - \beta_1 \operatorname{Re} g_{0021}(0)}{\operatorname{Re} g_{0021}(0) \operatorname{Re} g_{2100}(0) - \operatorname{Re} g_{1011}(0) \operatorname{Re} g_{1110}(0)}}, \sqrt{\frac{\beta_2 \operatorname{Re} g_{2100}(0) - \beta_1 \operatorname{Re} g_{1110}(0)}{\operatorname{Re} g_{1011}(0) \operatorname{Re} g_{1110}(0) - \operatorname{Re} g_{0021}(0) \operatorname{Re} g_{2100}(0)}} \right)$$

corresponds to a torus of the original system. When

$$\operatorname{Re}(g_{1110}(0))\beta_1 = \operatorname{Re}(g_{2100}(0))\beta_2$$

the nontrivial equilibrium coincides with the first semi-trivial equilibrium, thus giving a predictor for a Neimark-Sacker bifurcation curve. Similarly, when

$$\operatorname{Re}(g_{0021}(0))\beta_1 = \operatorname{Re}(g_{1011}(0))\beta_2$$

the nontrivial equilibrium coincides with the second semi-trivial equilibrium, and gives a predictor for the second Neimark-Sacker bifurcation curve. Therefore, we obtain two Neimark-Sacker bifurcation curves in (72), with approximations given by

$$(\rho_1, \rho_2, \beta_1, \beta_2) = (\epsilon, 0, -\operatorname{Re}(g_{2100}(0))\epsilon^2, -\operatorname{Re}(g_{1110}(0))\epsilon^2), \quad (74a)$$

$$(\rho_1, \rho_2, \beta_1, \beta_2) = (0, \epsilon, -\operatorname{Re}(g_{1011}(0))\epsilon^2, -\operatorname{Re}(g_{0021}(0))\epsilon^2), \quad (74b)$$

where $\epsilon > 0$, which are the predictors given in [46] and [49]. An approximation for the period of the cycle for each Neimark-Sacker predictor can be obtained from the third and fourth equation in system (73), yielding

$$\begin{cases} T_1 = 2\pi / (\omega_1 + b_{11}\beta_1 + b_{12}\beta_2 + \operatorname{Im}(g_{2100}(0))\epsilon^2), \\ T_2 = 2\pi / (\omega_2 + b_{21}\beta_1 + b_{22}\beta_2 + \operatorname{Im}(g_{0021}(0))\epsilon^2), \end{cases}$$

where

$$b_{jk} = \frac{\partial b_j}{\partial \beta_k}(0), \quad j, k = 1, 2.$$

Here we should use (β_1, β_2) as in (74a) and (74b).

5.4 Transcritical-Hopf bifurcation

A majority of papers in which fold-Hopf bifurcations in DDEs are studied, deals with models where the steady state remains fixed under variation of parameters. In this case the unfolding is not given by (67) anymore and we have to consider the smooth local normal form

$$\begin{cases} \dot{z}_0 = \gamma(\alpha)z_0 + g_{200}(\alpha)z_0^2 + g_{011}(\alpha)|z_1|^2 + g_{300}(\alpha)z_0^3 + g_{111}(\alpha)z_0|z_1|^2 \\ \quad + \mathcal{O}(\|(z_0, z_1, \bar{z}_1)\|^4), \\ \dot{z}_1 = \lambda(\alpha)z_1 + g_{110}(\alpha)z_0z_1 + g_{210}(\alpha)z_1^2 + g_{021}(\alpha)z_1|z_1|^2 + \mathcal{O}(\|(z_0, z_1, \bar{z}_1)\|^4). \end{cases}$$

The bifurcation analysis can be carried out similar to the fold-Hopf case, see [29] and [40]. An alternative approach is presented in [62]. In contrast with the fold-Hopf bifurcation, there are in general two Neimark-Sacker bifurcation curves. Furthermore, the fold bifurcation curve becomes a transcritical bifurcation curve, and meets the Hopf bifurcation curve transversally.

Under the assumption of the same transversality condition as in the fold-Hopf bifurcation we introduce new parameters $(\beta_1(\alpha), \beta_2(\alpha))$ to obtain the truncated normal form

$$\begin{cases} \dot{z}_0 = \beta_1 z_0 + g_{200}(\beta) z_0^2 + g_{011}(\beta) |z_1|^2 + g_{111}(\beta) z_0 |z_1|^2 + g_{300}(\beta) z_0^3, \\ \dot{z}_1 = (\beta_2 + i\omega_0 + ib_1(\beta)) z_1 + g_{110}(\beta) z_0 z_1 + g_{210}(\beta) z_0^2 z_1 + g_{021}(\beta) z_1 |z_1|^2, \end{cases} \quad (75)$$

with $b_1(0) = 0$. Letting $z_1 = \rho e^{i\psi}$ and separating the real and imaginary parts yields the three dimensional system

$$\begin{cases} \dot{z}_0 = \beta_1 z_0 + g_{200}(\beta) z_0^2 + g_{011}(\beta) \rho^2 + g_{111}(\beta) z_0 \rho^2 + g_{300}(\beta) z_0^3, \\ \dot{\rho} = \rho (\beta_2 + \operatorname{Re}(g_{110}(\beta)) z_0 + \operatorname{Re}(g_{210}(\beta)) z_0^2 + \operatorname{Re}(g_{021}(\beta)) \rho^2), \\ \dot{\psi} = \omega_0 + b_1(\beta) + \operatorname{Im}(g_{110}(\beta)) z_0 + \operatorname{Im}(g_{210}(\beta)) z_0^2 + \operatorname{Im}(g_{021}(\beta)) \rho^2. \end{cases} \quad (76)$$

5.4.1 Neimark-Sacker bifurcation curves

Following the same procedure as in [Section 5.2](#), we obtain that for $g_{011}(0) \operatorname{Re}(g_{110}(0)) < 0$ there are two Neimark-Sacker bifurcation curves approximated by

$$\begin{cases} \beta_1 = \mp 2\sqrt{g_{011}(0)g_{200}(0)}\epsilon, \\ \beta_2 = \mp \operatorname{Re}(g_{110}(0)) \sqrt{\frac{g_{011}(0)}{g_{200}(0)}}\epsilon, \\ z_0 = \pm \sqrt{\frac{g_{011}(0)}{g_{200}(0)}}\epsilon, \end{cases} \quad (77)$$

while the period of the corresponding cycle is approximated by

$$T = 2\pi / (\omega_0 + \omega_1 \beta_1 + \omega_2 \beta_2 + \operatorname{Im}(g_{110}(0)) z_0).$$

Here (z_0, β_1, β_2) are as in [\(77\)](#) and $\omega_i = \frac{\partial b_1}{\partial \beta_i}(0)$, for $i = 1, 2$.

5.4.2 Hopf and transcritical bifurcations curves

The transcritical bifurcation curve in the normal form is obtained by substituting $\rho = 0$ in the amplitude system of [\(76\)](#). Then β_2 is unrestricted and $z_0 = -\beta_1/g_{200}(0)$. The transcritical bifurcation curve is therefore given by

$$(\beta_1, \beta_2) = (0, \beta_2).$$

To obtain a predictor for the Hopf bifurcation curve we truncate [\(76\)](#) to the second order. We obtain a trivial equilibrium $(z_0, \rho) = (0, 0)$, a semi-trivial equilibrium $(z_0, \rho) = (-\frac{\beta_1}{g_{200}(0)}, 0)$ and a nontrivial equilibrium

$$(z_0, \rho) = \left(-\frac{\beta_2}{\operatorname{Re}(g_{110}(0))}, \frac{\sqrt{\beta_2 (\operatorname{Re}(g_{110}(0)) \beta_1 - g_{200}(0) \beta_2)}}{\operatorname{Re}(g_{110}(0)) \sqrt{g_{011}(0)}} \right).$$

It follows that the Hopf bifurcation curves are approximated by

$$\beta_2 = \frac{\operatorname{Re}(g_{110}(0))}{g_{200}(0)} \beta_1, \quad \beta_2 = 0.$$

6 Coefficients of parameter-dependent normal forms

Using the method outlined in [Section 4](#), we derive here the coefficients needed for the predictors of the nonhyperbolic equilibria and cycles emanating from generalized Hopf, fold-Hopf, Hopf-Hopf and transcritical-Hopf bifurcations, see [Section 5](#). While doing so, we also obtain the critical normal form coefficients, which were first derived in [\[37\]](#). Recall, that we consider only two-parameter DDEs, so that $p = 2$.

For the derivation of the coefficients in this section it is sufficient to expand and truncate the nonlinearity R and the parameter-mapping K in [\(57\)](#) and [\(58\)](#), respectively, as follows

$$R(u, \alpha) = \left(\frac{1}{2}B(u, u) + A_1(u, \alpha) + \frac{1}{2}J_2(\alpha, \alpha) + \frac{1}{6}C(u, u, u) + \frac{1}{2}B_1(u, u, \alpha) + \frac{1}{24}D(u, u, u, u) + \frac{1}{6}C_1(u, u, u, \alpha) + \frac{1}{120}E(u, u, u, u, u) \right) r^{\odot*}, \quad (78)$$

$$\alpha = K(\beta) = K_{10}\beta_1 + K_{01}\beta_2. \quad (79)$$

Here $u \in X$, while $\alpha, \beta \in \mathbb{R}^2$, and $B, A_1, J_2, C, B_1, D, C_1$ and E are the standard multilinear forms arising from the expansion of $F(u, \alpha)$ (or $G(u, \alpha)$). For example,

$$B(u, u) = D_1^2 F(0, 0)(u, u), \quad A_1(\alpha, \alpha) = D_2^2 F(0, 0)(\alpha, \alpha), \quad B_1(u, u, \alpha) = D_2^1 D_1^2 F(0, 0)(u, u, \alpha),$$

etc. These forms are \mathbb{R}^n -valued on real arguments, while they are linearly extended ('complexified') to \mathbb{C}^n -valued ones on complex-valued arguments. Finally, we introduce

$$J_1 = D_2 F(0, 0). \quad (80)$$

We assume in all situations that $\varphi_0 = 0$ is a steady state of [\(28\)](#) at the critical parameter value $\alpha_0 = 0 \in \mathbb{R}^2$. Explicit formulas to compute the multilinear forms for the simplest DDE [\(1\)](#) are given in [Section 7](#).

6.1 Generalized Hopf bifurcation

Since the eigenvalues [\(59\)](#) are simple, there exist eigenfunctions φ and φ^\odot such that

$$A\varphi = i\omega_0\varphi, \quad A^*\varphi^\odot = i\omega_0\varphi^\odot, \quad \langle \varphi^\odot, \varphi \rangle = 1.$$

The eigenfunctions φ and φ^\odot are explicitly given by [\(22\)](#) and [\(23\)](#) with $q \in \mathbb{C}^n$ and $p \in \mathbb{C}^{n*}$ satisfying

$$\Delta(i\omega_0)q = 0, \quad p\Delta(i\omega_0) = 0, \quad p\Delta'(i\omega_0)q = 1.$$

Any point $y \in X_0$ in the real critical eigenspace can be represented as

$$y = z\varphi + \bar{z}\bar{\varphi}, \quad z \in \mathbb{C},$$

where $z = \langle \varphi^\odot, y \rangle$. Therefore, the homological equation [\(HOM\)](#) can be written as

$$A^{\odot*}j\mathcal{H}(z, \bar{z}, \beta) + J_1K(\beta)r^{\odot*} + R(\mathcal{H}(z, \bar{z}, \beta), K(\beta)) = j(D_z\mathcal{H}(z, \bar{z}, \beta)\dot{z} + D_{\bar{z}}\mathcal{H}(z, \bar{z}, \beta)\dot{\bar{z}}),$$

where $\beta = (\beta_1, \beta_2)$, \dot{z} is given by the normal form [\(61\)](#) and \mathcal{H} admits the expansion

$$\mathcal{H}(z, \bar{z}, \beta) = z\varphi + \bar{z}\bar{\varphi} + H_{0010}\beta_1 + H_{0001}\beta_2 + \sum_{j+k+|\mu| \geq 2} \frac{1}{j!k!\mu!} H_{jk\mu} z^j \bar{z}^k \beta^\mu. \quad (81)$$

For the predictors derived in [Section 5.1](#) we need the truncated parameter-dependent normal form

$$\dot{z} = (i\omega_0 + \beta_1 + i\omega_{10}\beta_1 + i\omega_{01}\beta_2)z + (\beta_2 + c_1(0))z|z|^2 + c_2(0)z|z|^4,$$

where $\text{Re}(c_1(0)) = 0$.

6.1.1 Critical normal form coefficients

We start by calculating the critical normal form coefficients following [37]. Collecting the coefficients of the quadratic terms z^2 and $z\bar{z}$ in the homological equation yields two nonsingular linear systems:

$$\begin{aligned} (2i\omega_0 - A^{\odot\star}) jH_{2000} &= B(\varphi, \varphi)r^{\odot\star}, \\ -A^{\odot\star} jH_{1100} &= B(\varphi, \bar{\varphi})r^{\odot\star}. \end{aligned}$$

They are solved using [Lemma 3](#) to give

$$\begin{aligned} H_{2000}(\theta) &= e^{2i\omega_0\theta} \Delta^{-1}(2i\omega_0)B(\varphi, \varphi), \\ H_{1100}(\theta) &= \Delta^{-1}(0)B(\varphi, \bar{\varphi}). \end{aligned}$$

For the cubic terms, the system corresponding to z^3 is also nonsingular,

$$(3i\omega_0 - A^{\odot\star}) jH_{3000} = [3B(\varphi, H_{2000}) + C(\varphi, \varphi, \varphi)] r^{\odot\star},$$

with solution

$$H_{3000}(\theta) = e^{3i\omega_0\theta} \Delta^{-1}(3i\omega_0) (3B(\varphi, H_{2000}) + C(\varphi, \varphi, \varphi)).$$

On the other hand, the system corresponding to $z^2\bar{z}$ is singular,

$$(i\omega_0 - A^{\odot\star}) jH_{2100} = [B(\bar{\varphi}, H_{2000}) + 2B(\varphi, H_{1100}) + C(\varphi, \varphi, \bar{\varphi})] r^{\odot\star} - 2c_1(0)j\varphi.$$

The Fredholm solvability condition ([FSC](#)) requires that

$$c_1(0) = \frac{1}{2}p \cdot (B(\bar{\varphi}, H_{2000}) + 2B(\varphi, H_{1100}) + C(\varphi, \varphi, \bar{\varphi})),$$

and from [Lemma 4](#) we then obtain the unique solution satisfying $\langle \varphi^{\odot}, H_{2100} \rangle = 0$ as

$$H_{2100}(\theta) = B_{i\omega_0}^{\text{INV}} (B(\bar{\varphi}, H_{2000}) + 2B(\varphi, H_{1100}) + C(\varphi, \varphi, \bar{\varphi}), -2c_1(0)) (\theta).$$

We continue by collecting the coefficients corresponding to the fourth-order terms $z^3\bar{z}$ and z^2z^2 in the homological equation. The corresponding nonsingular systems may be solved using [Lemma 3](#) and the fact that $\text{Re}(c_1(0)) = 0$. For H_{2200} this easily gives

$$\begin{aligned} H_{2200}(\theta) &= \Delta^{-1}(0)[2B(\bar{\varphi}, H_{2100}) + 2B(\varphi, \bar{H}_{2100}) + B(\bar{H}_{2000}, H_{2000}) \\ &\quad + 2B(H_{1100}, H_{1100}) + C(\varphi, \varphi, \bar{H}_{2000}) + 4C(\varphi, \bar{\varphi}, H_{1100}) \\ &\quad + C(\bar{\varphi}, \bar{\varphi}, H_{2000}) + D(\varphi, \varphi, \bar{\varphi}, \bar{\varphi})], \end{aligned}$$

but for H_{3100} the solution is a bit more subtle. The linear system is

$$\begin{aligned} (2i\omega_0 - A^{\odot\star}) jH_{3100} &= [B(\bar{\varphi}, H_{3000}) + 3B(\varphi, H_{2100}) + 3B(H_{1100}, H_{2000}) + 3C(\varphi, \bar{\varphi}, H_{2000}) \\ &\quad + 3C(\varphi, \varphi, H_{1100}) + D(\varphi, \varphi, \varphi, \bar{\varphi})] r^{\odot\star} - 6c_1(0)jH_{2000}, \end{aligned}$$

so [Lemma 3](#) applies with $w_0 = [\dots] - 6c_1(0)H_{2000}(0)$ and $w = -6c_1(0)H_{2000}$ and we find

$$\begin{aligned} H_{3100}(\theta) &= e^{2i\omega_0\theta} \Delta^{-1}(2i\omega_0)[B(\bar{\varphi}, H_{3000}) + 3B(\varphi, H_{2100}) + 3B(H_{1100}, H_{2000}) \\ &\quad + 3C(\varphi, \bar{\varphi}, H_{2000}) + 3C(\varphi, \varphi, H_{1100}) + D(\varphi, \varphi, \varphi, \bar{\varphi})] \\ &\quad - 6c_1(0)\Delta^{-1}(2i\omega_0)[\Delta'(2i\omega_0) - \theta\Delta(2i\omega_0)]H_{2000}(\theta). \end{aligned}$$

The critical normal form coefficient $c_2(0)$ is calculated by applying (FSC) to the singular linear system corresponding to the fifth-order term $z^3\bar{z}^2$ in the homological equation. This gives

$$\begin{aligned} c_2(0) = & \frac{1}{12}p \cdot [2B(\bar{\varphi}, H_{3100}) + 3B(\varphi, H_{2200}) + B(\bar{H}_{2000}, H_{3000}) \\ & + 6B(H_{1100}, H_{2100}) + 3B(\bar{H}_{2100}, H_{2000}) \\ & + 6C(\bar{\varphi}, H_{1100}, H_{2000}) + 6C(\varphi, \bar{\varphi}, H_{2100}) + C(\bar{\varphi}, \bar{\varphi}, H_{3000}) \\ & + 3C(\varphi, \varphi, \bar{H}_{2100}) + 3C(\varphi, \bar{H}_{2000}, H_{2000}) + 6C(\varphi, H_{1100}, H_{1100}) \\ & + D(\varphi, \varphi, \varphi, \bar{H}_{2000}) + 6D(\varphi, \varphi, \bar{\varphi}, H_{1100}) + 3D(\varphi, \bar{\varphi}, \bar{\varphi}, H_{2000}) \\ & + E(\varphi, \varphi, \varphi, \bar{\varphi}, \bar{\varphi})]. \end{aligned}$$

The second Lyapunov coefficient is now given by $\ell_2(0) = \frac{1}{\omega_0} \operatorname{Re}(c_2(0))$.

6.1.2 Parameter-related coefficients

Next we derive the parameter-related coefficients that provide a linear approximation to the parameter transformation. Following [46] and [2] we first expand the eigenvalue $\lambda(\alpha)$ and $c_1(\alpha)$ in the normal form (60) in the original parameters α and truncate to fourth order,

$$\dot{z} = (i\omega_0 + \gamma_{110}\alpha_1 + \gamma_{101}\alpha_2)z + (c_1(0) + \gamma_{210}\alpha_1 + \gamma_{201}\alpha_2)z|z|^2 + c_2(0)z|z|^4.$$

The parameters α and β are related via

$$\alpha = \left(\operatorname{Re} \begin{pmatrix} \gamma_{110} & \gamma_{101} \\ \gamma_{210} & \gamma_{201} \end{pmatrix} \right)^{-1} \beta \quad (82)$$

so that

$$\frac{\partial b_1}{\partial \beta_2}(0) = \operatorname{Im} \left(\begin{pmatrix} \gamma_{110} & \gamma_{101} \end{pmatrix} \left(\operatorname{Re} \begin{pmatrix} \gamma_{110} & \gamma_{101} \\ \gamma_{210} & \gamma_{201} \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right). \quad (83)$$

The homological equation (HOM) becomes

$$A^{\odot\star} j\mathcal{H}(z, \bar{z}, \alpha) + J_1 \alpha r^{\odot\star} + R(\mathcal{H}(z, \bar{z}, \alpha), \alpha) = j(D_z \mathcal{H}(z, \bar{z}, \alpha)\dot{z} + D_{\bar{z}} \mathcal{H}(z, \bar{z}, \alpha)\dot{\bar{z}}),$$

where \mathcal{H} admits the expansion

$$u = \mathcal{H}(z, \bar{z}, \alpha_1, \alpha_2) = z\varphi + \bar{z}\bar{\varphi} + H_{0010}\alpha_1 + H_{0001}\alpha_2 + \sum_{j+k+|\mu|\geq 2} \frac{1}{j!k!\mu!} H_{jk\mu} z^j \bar{z}^k \alpha^\mu \quad (84)$$

and R is given by (78). Notice that the coefficients $H_{jk\mu}$ with $\mu = (00)$ in (84) coincide with those in the expansion (81). Collecting the coefficients of the terms α and $z\alpha$ in the homological equation yields the systems

$$\begin{aligned} -A^{\odot\star} jH_{00\mu} &= J_1 v_\mu r^{\odot\star}, \\ (i\omega_0 - A^{\odot\star}) jH_{10\mu} &= [A_1(\varphi, v_\mu) + B(\varphi, H_{00\mu})] r^{\odot\star} - \gamma_{1\mu} j\varphi, \end{aligned}$$

where $\mu = (10), (01)$ and $v_{10} = (1, 0)^T, v_{01} = (0, 1)^T$. We solve these systems using Section 2.6. By the first part of Lemma 3 the first system has the (constant) solutions

$$H_{00\mu}(\theta) = \Delta^{-1}(0) J_1 v_\mu$$

and (FSC) gives

$$\gamma_{1\mu} = p(A_1(\varphi, v_\mu) + B(\varphi, H_{00\mu})).$$

Using [Lemma 4](#) we obtain the solutions

$$H_{10\mu}(\theta) = B_{i\omega_0}^{\text{INV}}(A_1(\varphi, v_\mu) + B(\varphi, H_{00\mu}), -\gamma_{1\mu})(\theta)$$

for the second equation. To determine $\gamma_{2\mu}$ we first collect the coefficients corresponding to the $z^2\alpha$ and $z\bar{z}\alpha$ terms in the homological equation. We obtain the equations

$$\begin{aligned} (2i\omega_0 - A^{\odot*})jH_{20\mu} &= [A_1(H_{2000}, v_\mu) + 2B(\varphi, H_{10\mu}) + B(H_{2000}, H_{00\mu}) + B_1(\varphi, \varphi, v_\mu) \\ &\quad + C(\varphi, \varphi, H_{00\mu})]r^{\odot*} - 2\gamma_{1\mu}jH_{2000}, \\ -A^{\odot*}jH_{11\mu} &= [A_1(H_{1100}, v_\mu) + 2\text{Re}(B(\bar{\varphi}, H_{10\mu})) + B(H_{1100}, H_{00\mu}) + B_1(\varphi, \bar{\varphi}, v_\mu) \\ &\quad + C(\varphi, \bar{\varphi}, H_{0\mu})]r^{\odot*} - 2\text{Re}(\gamma_{1\mu})jH_{1100}. \end{aligned}$$

[Lemma 3](#) implies that solutions of the first two equations are given by

$$\begin{aligned} H_{20\mu}(\theta) &= e^{2i\omega_0\theta}\Delta^{-1}(2i\omega_0)[A_1(H_{2000}, v_\mu) + 2B(\varphi, H_{10\mu}) + B(H_{2000}, H_{00\mu}) + B_1(\varphi, \varphi, v_\mu) \\ &\quad + C(\varphi, \varphi, H_{00\mu})] - 2\gamma_{1\mu}\Delta(2i\omega_0)^{-1}(\Delta'(2i\omega) - \theta\Delta(2i\omega_0))H_{2000}(\theta), \\ H_{11\mu}(\theta) &= \Delta^{-1}(0)[A_1(H_{1100}, v_\mu) + 2\text{Re}(B(\bar{\varphi}, H_{10\mu})) + B(H_{1100}, H_{00\mu}) + B_1(\varphi, \bar{\varphi}, v_\mu) \\ &\quad + C(\varphi, \bar{\varphi}, H_{0\mu})] - 2\text{Re}(\gamma_{1\mu})\Delta(0)^{-1}(\Delta'(0) - \theta\Delta(0))H_{1100}(\theta). \end{aligned}$$

Applying [\(FSC\)](#) to $z^2\bar{z}\alpha$ terms in the homological equation results in

$$\begin{aligned} \gamma_{2\mu} &= \frac{1}{2}p \cdot [A_1(H_{2100}, v_\mu) + B(\bar{\varphi}, H_{20\mu}) + 2B(\varphi, H_{11\mu}) \\ &\quad + B(H_{2100}, H_{00\mu}) + B(H_{2000}, \bar{H}_{10\mu}) + 2B(H_{1100}, H_{10\mu}) \\ &\quad + B_1(H_{2000}, \bar{\varphi}, v_\mu) + 2B_1(\varphi, H_{1100}, v_\mu) + 2C(\varphi, \bar{\varphi}, H_{10\mu}) \\ &\quad + C(H_{2000}, \bar{\varphi}, H_{00\mu}) + C(\varphi, \varphi, H_{01\mu}) + 2C(\varphi, H_{1100}, H_{00\mu}) \\ &\quad + C_1(\varphi, \varphi, \bar{\varphi}, v_\mu) + D(\varphi, \varphi, \bar{\varphi}, H_{00\mu})]. \end{aligned}$$

6.1.3 Hopf and LPC predictors

Now we are ready to specify the predictors for the original parameter-dependent DDE [\(28\)](#). To approximate the Hopf parameter values α and the corresponding equilibrium, we merely substitute β from [\(65\)](#) into [\(82\)](#), and then put the result together with $z = 0$ into the expansion [\(84\)](#).

To approximate the LPC parameter values, we substitute β from [\(63\)](#) into [\(82\)](#). The cycle period is approximated by [\(64\)](#) with [\(83\)](#). To obtain a predictor for the periodic orbit in the phase space, we set $z = \epsilon e^{i\psi}$ into [\(84\)](#) using the obtained α values. Truncating to the second order in ϵ then yields

$$u = 2\text{Re}(e^{i\psi}\varphi)\epsilon + (H_{1100} - 2\text{Re}(c_2(0))H_{0001} + \text{Re}(e^{2i\psi}H_{2000}))\epsilon^2, \quad \psi \in [0, 2\pi].$$

6.2 Fold-Hopf bifurcation

Since the eigenvalues [\(66\)](#) are simple, there exist eigenfunctions $\varphi_{0,1}$ and $\varphi_{0,1}^{\odot}$ satisfying

$$A\varphi_0 = 0, \quad A\varphi_1 = i\omega_0\varphi_1, \quad A^*\varphi_0^{\odot} = 0, \quad A^*\varphi_1^{\odot} = i\omega_0\varphi_1^{\odot},$$

as well as the mutual normalization condition

$$\langle \varphi_i^{\odot}, \varphi_j \rangle = \delta_{ij}, \quad 0 \leq i, j \leq 1.$$

The eigenfunctions $\varphi_{0,1}$ and $\varphi_{0,1}^{\odot}$ can be explicitly computed using [\(22\)](#) and [\(23\)](#) with $q_0 \in \mathbb{R}^n$, $q_1 \in \mathbb{C}^n$, $p_0 \in \mathbb{R}^{n^*}$ and $p_1 \in \mathbb{C}^{n^*}$ satisfying

$$\Delta(0)q_0 = 0, \quad \Delta(i\omega_0)q_1 = 0, \quad p_0\Delta(0) = 0, \quad p_1\Delta(i\omega_0) = 0,$$

as well as

$$p_0 \Delta'(0)q_0 = 1, \quad p_1 \Delta'(i\omega_0)q_1 = 1.$$

Any point $y \in X_0$ in the real critical eigenspace can be represented as

$$y = z_0 \varphi_0 + z_1 \varphi_1 + \bar{z}_1 \bar{\varphi}_1, \quad (z_0, z_1) \in \mathbb{R} \times \mathbb{C},$$

where $z_0 = \langle \varphi_0^\odot, y \rangle$ and $z_1 = \langle \varphi_1^\odot, y \rangle$. Therefore, the homological equation (HOM) can be written as

$$\begin{aligned} A^{\odot*} j \mathcal{H}(z, \beta) + J_1(\beta) r^{\odot*} + R(\mathcal{H}(z, \beta), K(\beta)) \\ = j(D_{z_0} \mathcal{H}(z, \beta) \dot{z}_0 + D_{z_1} \mathcal{H}(z, \beta) \dot{z}_1 + D_{\bar{z}_1} \mathcal{H}(z, \beta) \dot{\bar{z}}_1), \end{aligned} \quad (85)$$

where $z = (z_0, z_1, \bar{z}_1)$, $\beta = (\beta_1, \beta_2)$ and \dot{z} is given by the normal form (68). Here, the mapping \mathcal{H} admits the expansion

$$\begin{aligned} \mathcal{H}(z_0, z_1, \bar{z}_1, \beta) &= z_0 \varphi_0 + z_1 \varphi_1 + \bar{z}_1 \bar{\varphi}_1 + H_{00010} \beta_1 + H_{00001} \beta_2 \\ &+ \sum_{j+k+l+|\mu| \geq 2} \frac{1}{j!k!l!\mu!} H_{jkl\mu} z_0^j z_1^k \bar{z}_1^l \beta^\mu, \end{aligned} \quad (86)$$

and the functions K and R are as in (78) and (79), respectively.

6.2.1 Critical normal form coefficients

We start by computing the critical normal form coefficients following [37]. Collecting the quadratic terms z_0^2 , z_1^2 , $z_0 z_1$ and $z_1 \bar{z}_1$ we obtain one nonsingular and three singular linear systems. By (FSC) the singular systems are consistent if and only if

$$g_{200}(0) = \frac{1}{2} p_0 B(\varphi_0, \varphi_0), \quad g_{110}(0) = p_1 B(\varphi_0, \varphi_1), \quad g_{011}(0) = p_0 B(\varphi_1, \bar{\varphi}_1).$$

This yields the three quadratic normal form coefficients. The corresponding solutions may then be obtained using Lemmas 3 and 4. Namely,

$$\begin{aligned} H_{20000}(\theta) &= B_0^{\text{INV}}(B(\varphi_0, \varphi_0), -2g_{200}(0))(\theta), \\ H_{02000}(\theta) &= e^{2i\omega_0 \theta} \Delta^{-1}(2i\omega_0) B(\varphi_1, \varphi_1), \\ H_{11000}(\theta) &= B_{i\omega_0}^{\text{INV}}(B(\varphi_0, \varphi_1), -g_{110}(0))(\theta), \\ H_{01100}(\theta) &= B_0^{\text{INV}}(B(\varphi_1, \bar{\varphi}_1), -g_{011}(0))(\theta). \end{aligned}$$

For the four remaining cubic normal form coefficients, we collect the coefficients of the resonant terms $z_0^j z_1^k \bar{z}_1^l$ in (85) with $j + k + l = 3$. This yields four singular linear systems. As before, by (FSC) these systems are consistent if and only if

$$\begin{aligned} g_{300}(0) &= \frac{1}{6} p_0 (3B(\varphi_0, H_{20000}) + C(\varphi_0, \varphi_0, \varphi_0)), \\ g_{111}(0) &= p_0 (B(\varphi_0, H_{01100}) + B(\varphi_1, \bar{H}_{11000}) + B(\bar{\varphi}_1, H_{11000}) + C(\varphi_0, \varphi_1, \bar{\varphi}_1)), \\ g_{210}(0) &= \frac{1}{2} p_1 (2B(\varphi_0, H_{11000}) + B(\varphi_1, H_{20000}) + C(\varphi_0, \varphi_0, \varphi_1)), \\ g_{021}(0) &= \frac{1}{2} p_1 (2B(\varphi_1, H_{01100}) + B(\bar{\varphi}_1, H_{02000}) + C(\varphi_1, \varphi_1, \bar{\varphi}_1)). \end{aligned}$$

6.2.2 Parameter-related coefficients

The parameter-related linear terms in (85) give

$$\begin{aligned} -A^{\odot*} j H_{00010} &= J_1 K_{10} r^{\odot*} - j \varphi_0, \\ -A^{\odot*} j H_{00001} &= J_1 K_{01} r^{\odot*}. \end{aligned}$$

Let $\gamma = (\gamma_1, \gamma_2) = p_0^T J_1$. Then by (FSC) we obtain the orthogonal frame

$$K_{10} = s_1 + \delta_1 s_2, \quad K_{01} = \delta_2 s_2, \quad (87)$$

where

$$s_1^T = \gamma / \|\gamma\|^2, \quad s_2^T = (-\gamma_2, \gamma_1)$$

and $\delta_{1,2} \in \mathbb{R}$ are constants. Using Lemma 4 from Section 2.6 we get

$$\begin{aligned} H_{00010}(\theta) &= \Delta^{\text{INV}}(0) (J_1 K_{10} - \Delta'(0) q_0) + \delta_3 q_0 + \theta q_0 \\ &= r_1 + \delta_1 r_2 + \delta_3 q_0 - r_3(\theta), \\ H_{00001}(\theta) &= \delta_2 r_2 + \delta_4 q_0, \end{aligned} \quad (88)$$

where

$$r_1 = \Delta^{\text{INV}}(0) (J_1 s_1), \quad r_2 = \Delta^{\text{INV}}(0) (J_1 s_2), \quad r_3(\theta) = \Delta^{\text{INV}}(0) (\Delta'(0) q_0) - \theta q_0,$$

and the real constants δ_3 and δ_4 are not chosen such that $\langle \varphi_0^{\odot}, H_{00010} \rangle = 0$ and $\langle \varphi_0^{\odot}, H_{00001} \rangle = 0$, but will be determined below. Collecting the $z_0\beta$ and $z_1\beta$ terms in the homological equation yields the systems

$$\begin{aligned} -A^{\odot*} j H_{10010} &= [B(\varphi_0, H_{00010}) + A_1(\varphi_0, K_{10})] r^{\odot*} - j H_{20000}, \\ -A^{\odot*} j H_{10001} &= [B(\varphi_0, H_{00001}) + A_1(\varphi_0, K_{01})] r^{\odot*}, \\ (i\omega_0 - A^{\odot*}) j H_{01010} &= [B(\varphi_1, H_{00010}) + A_1(\varphi_1, K_{10})] r^{\odot*} - j (i\omega_1 \varphi_1 + H_{11000}), \\ (i\omega_0 - A^{\odot*}) j H_{01001} &= [B(\varphi_1, H_{00001}) + A_1(\varphi_1, K_{01})] r^{\odot*} - (1 + i\omega_2) j \varphi_1. \end{aligned} \quad (89)$$

Notice that the coefficients $\omega_{1,2}$ were introduced in Section 5.2.1. To determine $\delta_i (i = 1, 2, 3, 4)$ we substitute (87) and (88) into (89). Then by (FSC) we obtain the system

$$\begin{pmatrix} p_0 B(\varphi_0, r_2) + p_0 A_1(\varphi_0, s_2) & 2g_{200}(0) \\ \text{Re}(p_1 B(\varphi_1, r_2) + p_1 A_1(\varphi_1, s_2)) & \text{Re}(g_{110}(0)) \end{pmatrix} \begin{pmatrix} \delta_1 & \delta_2 \\ \delta_3 & \delta_4 \end{pmatrix} = \begin{pmatrix} -p_0 (A_1(\varphi_0, s_1) + B(\varphi_0, r_1 - r_3)) & 0 \\ -\text{Re}(p_1 (A_1(\varphi_1, s_1) + B(\varphi_1, r_1 - r_3))) & 1 \end{pmatrix}.$$

Subsequently, the coefficients ω_1 and ω_2 are given by

$$\begin{aligned} \omega_1 &= \text{Im} (p_1 B(\varphi_1, H_{00010}) + p_1 A_1(\varphi_1, K_{10})), \\ \omega_2 &= \text{Im} (p_1 B(\varphi_1, H_{00001}) + p_1 A_1(\varphi_1, K_{01})). \end{aligned}$$

6.2.3 Hopf, fold, and Neimark-Sacker predictors

To approximate the fold and Hopf curves and their corresponding equilibria, one should substitute the expressions for β and the equilibrium coordinates given in Section 5.2.2 into the expansions (86) and (79).

To approximate the periodic orbit at the Neimark-Sacker bifurcation, we substitute $z_1 = \epsilon e^{i\psi}$ and (70) into (86). After a truncation this gives

$$u = 2 \operatorname{Re}(e^{i\psi} \varphi_1) \epsilon + \left(\frac{\operatorname{Re}(g_{110}(0)) (2 \operatorname{Re}(g_{021}(0)) + g_{111}(0)) - 2 \operatorname{Re}(g_{021}(0)) g_{200}(0)}{2g_{200}(0)} H_{00001} - g_{011}(0) H_{00010} + H_{01100} - \left(\frac{2 \operatorname{Re}(g_{021}(0)) + g_{111}(0)}{2g_{200}(0)} \right) \varphi_0 + \operatorname{Re}(e^{2i\psi} \bar{H}_{02000}) \right) \epsilon^2,$$

where $\psi \in [0, 2\pi]$.

6.3 Hopf-Hopf bifurcation

Since the eigenvalues (71) are simple, there exist eigenfunctions $\varphi_{1,2}$ and $\varphi_{1,2}^\circ$,

$$A\varphi_1 = i\omega_1\varphi_1, \quad A\varphi_2 = i\omega_2\varphi_2, \quad A^*\varphi_1^\circ = i\omega_1\varphi_1^\circ, \quad A^*\varphi_2^\circ = i\omega_2\varphi_2^\circ, \quad (90)$$

satisfying the mutual normalization conditions

$$\langle \varphi_i^\circ, \varphi_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq 2.$$

The eigenfunctions $\varphi_{1,2}$ and $\varphi_{1,2}^\circ$ can be explicitly computed using (22) and (23) with $q_{1,2} \in \mathbb{C}^n$ and $p_{1,2} \in \mathbb{C}^{n*}$ such that both

$$\Delta(i\omega_1)q_1 = 0, \quad \Delta(i\omega_2)q_2 = 0, \quad p_1\Delta(i\omega_1) = 0, \quad p_2\Delta(i\omega_2) = 0,$$

as well as

$$p_1\Delta'(i\omega_1)q_1 = 1, \quad p_2\Delta'(i\omega_2)q_2 = 1.$$

Any point $y \in X_0$ in the real critical eigenspace can be represented as

$$y = z_1\varphi_1 + \bar{z}_1\bar{\varphi}_1 + z_2\varphi_2 + \bar{z}_2\bar{\varphi}_2, \quad z_{1,2} \in \mathbb{C},$$

where $z_1 = \langle \varphi_1^\circ, y \rangle$ and $z_2 = \langle \varphi_2^\circ, y \rangle$. Therefore, the homological equation (HOM) can be written as

$$A^{\circ*}\mathcal{H}(z, \beta) + J_1(\beta)r^{\circ*} + R(\mathcal{H}(z, \beta), K(\beta)) = j(D_{z_1}\mathcal{H}(z, \beta)\dot{z}_1 + D_{\bar{z}_1}\mathcal{H}(z, \beta)\dot{\bar{z}}_1 + D_{z_2}\mathcal{H}(z, \beta)\dot{z}_2 + D_{\bar{z}_2}\mathcal{H}(z, \beta)\dot{\bar{z}}_2), \quad (91)$$

where $z = (z_1, \bar{z}_1, z_2, \bar{z}_2)$, $\beta = (\beta_1, \beta_2)$ and \dot{z} is given by the normal form (72). The mapping \mathcal{H} admits the expansion

$$\begin{aligned} \mathcal{H}(z_1, \bar{z}_1, z_2, \bar{z}_2, \beta_1, \beta_2) &= z_1\varphi_1 + \bar{z}_1\bar{\varphi}_1 + z_2\varphi_2 + \bar{z}_2\bar{\varphi}_2 + H_{000010}\beta_1 + H_{000001}\beta_2 \\ &+ \sum_{j+k+l+m+|\mu| \geq 2} \frac{1}{j!k!l!m!\mu!} H_{jklm\mu} z_1^j \bar{z}_1^k z_2^l \bar{z}_2^m \beta^\mu \end{aligned} \quad (92)$$

and the functions K and R are as in (78) and (79), respectively.

6.3.1 Critical normal form coefficients

For initialization of the Neimark-Sacker curves (74) we need the cubic critical normal form coefficients $g_{2100}(0)$, $g_{1011}(0)$, $g_{1110}(0)$ and $g_{0021}(0)$. We compute these coefficients following [37].

Collecting the coefficients of the quadratic terms $|z_1|^2$, z_1^2 , $z_1 z_2$, $|z_2|^2$, $z_1 \bar{z}_2$ and $z_2 \bar{z}_1$ in the homological equation, we obtain six nonsingular linear systems. By [Lemma 3](#) their solutions are

$$\begin{aligned} H_{110000}(\theta) &= \Delta^{-1}(0)B(\varphi_1, \bar{\varphi}_1), \\ H_{200000}(\theta) &= e^{2i\omega_1\theta} \Delta^{-1}(2i\omega_1)B(\varphi_1, \varphi_1), \\ H_{101000}(\theta) &= e^{i(\omega_1+\omega_2)\theta} \Delta^{-1}(i(\omega_1+\omega_2))B(\varphi_1, \varphi_2), \\ H_{001100}(\theta) &= \Delta^{-1}(0)B(\varphi_2, \bar{\varphi}_2), \\ H_{100100}(\theta) &= e^{i(\omega_1-\omega_2)\theta} \Delta^{-1}(i(\omega_1-\omega_2))B(\varphi_1, \bar{\varphi}_2), \\ H_{002000}(\theta) &= e^{2i\omega_2\theta} \Delta^{-1}(2i\omega_2)B(\varphi_2, \varphi_2). \end{aligned}$$

The desired cubic critical normal form coefficients are obtained by collecting the coefficients of the resonant cubic terms $z_1|z_1|^2$, $z_1|z_2|^2$, $|z_1|^2 z_2$ and $|z_2|^2 z_2$ in the homological equation. This leads to four singular linear systems. By [\(FSC\)](#) these systems are solvable if and only if

$$\begin{aligned} g_{2100}(0) &= \frac{1}{2}p_1 (2B(\varphi_1, H_{110000}) + B(\bar{\varphi}_1, H_{200000}) + C(\varphi_1, \varphi_1, \bar{\varphi}_1)), \\ g_{1011}(0) &= p_1 (B(\bar{\varphi}_2, H_{101000}) + B(\varphi_1, H_{001100}) + B(\varphi_2, H_{100100}) + C(\varphi_1, \varphi_2, \bar{\varphi}_2)), \\ g_{1110}(0) &= p_2 (B(\bar{\varphi}_1, H_{101000}) + B(\varphi_1, \bar{H}_{100100}) + B(\varphi_2, H_{110000}) + C(\varphi_1, \bar{\varphi}_1, \varphi_2)), \\ g_{0021}(0) &= \frac{1}{2}p_2 (2B(\varphi_2, H_{001100}) + B(\bar{\varphi}_2, H_{002000}) + C(\varphi_2, \varphi_2, \bar{\varphi}_2)). \end{aligned}$$

6.3.2 Parameter-related coefficients

The linear terms in [\(91\)](#) give back the eigenfunctions [\(90\)](#) and the parameter-related equations

$$-A^{\odot\star} j H_{0000\mu} = J_1 K_\mu r^{\odot\star},$$

where $\mu = (10), (01)$. Let

$$K_\mu = \gamma_{1\mu} e_1 + \gamma_{2\mu} e_2, \quad (93)$$

where $e_1 = (1, 0)$, $e_2 = (0, 1)$ and $\gamma_{i\mu} (i = 1, 2) \in \mathbb{R}$ are constants to be determined. Then [Lemma 3](#) from [Section 2.6](#) implies

$$H_{0000\mu}(\theta) = \gamma_{1\mu} \Delta^{-1}(0) J_1 e_1 + \gamma_{2\mu} \Delta^{-1}(0) J_1 e_2. \quad (94)$$

Collecting the $z_i \beta_j$ -terms with $1 \leq i, j \leq 2$ yields the systems

$$\begin{aligned} (i\omega_1 - A^{\odot\star}) j H_{100010} &= [A_1(\varphi_1, K_{10}) + B(\varphi_1, H_{000010})] r^{\odot\star} - (1 + ib_{11}) j \varphi_1, \\ (i\omega_1 - A^{\odot\star}) j H_{100001} &= [A_1(\varphi_1, K_{01}) + B(\varphi_1, H_{000001})] r^{\odot\star} - ib_{12} j \varphi_1, \\ (i\omega_2 - A^{\odot\star}) j H_{001010} &= [A_1(\varphi_2, K_{10}) + B(\varphi_2, H_{000010})] r^{\odot\star} - ib_{21} j \varphi_2, \\ (i\omega_2 - A^{\odot\star}) j H_{001001} &= [A_1(\varphi_2, K_{01}) + B(\varphi_2, H_{000001})] r^{\odot\star} - (1 + ib_{22}) j \varphi_2, \end{aligned} \quad (95)$$

where b_{jk} are defined in [Section 5.3.2](#). To determine $\gamma_{i\mu} (i = 1, 2)$ we substitute [\(93\)](#) and [\(94\)](#) into [\(95\)](#). Then by [\(FSC\)](#) we obtain the system

$$\operatorname{Re} \left[\begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{31} & \Gamma_{32} \end{pmatrix} \right] \begin{pmatrix} \gamma_{110} & \gamma_{210} \\ \gamma_{101} & \gamma_{201} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

where

$$\Gamma_{ij} := A_1(\varphi_i, e_j) + B(\varphi_i, \Delta^{-1}(0) J_1 e_j), \quad 1 \leq i, j \leq 2.$$

Note that $\Delta^{-1}(0) J_1 e_i$ is a constant function of θ .

It now follows from (95) that the coefficients b_{11}, b_{12}, b_{21} and b_{22} , needed for the second order approximation of the periods, are given by

$$\begin{aligned} b_{11} &= \text{Im}(p_1(A_1(\varphi_1, K_{10}) + B(\varphi_1, H_{000010}))), \\ b_{12} &= \text{Im}(p_1(A_1(\varphi_1, K_{01}) + B(\varphi_1, H_{000001}))), \\ b_{21} &= \text{Im}(p_2(A_1(\varphi_2, K_{10}) + B(\varphi_2, H_{000010}))), \\ b_{22} &= \text{Im}(p_2(A_1(\varphi_2, K_{01}) + B(\varphi_2, H_{000001}))). \end{aligned}$$

6.3.3 Hopf and Neimark-Sacker predictors

To approximate the Hopf curves and their corresponding equilibria, one should substitute the expressions for β and the equilibrium coordinates given in Section 5.3.1 into the expansions (92) and (79).

To approximate the Neimark-Sacker periodic orbits, we substitute $(z_1, z_2) = (\epsilon e^{i\psi_1}, 0)$ and (74a), and $(z_1, z_2) = (0, \epsilon e^{i\psi_2})$ and (74b) into (92). After a truncation, we obtain

$$\begin{aligned} u_1 &= 2 \text{Re}(e^{i\psi_1} \varphi_1) \epsilon + \left(-\text{Re}(g_{1110}(0))H_{000001} - \text{Re}(g_{2100}(0))H_{000010} \right. \\ &\quad \left. + H_{110000} + \text{Re}(e^{2i\psi_1} H_{200000}) \right) \epsilon^2, \quad \psi_1 \in [0, 2\pi] \end{aligned}$$

and

$$\begin{aligned} u_2 &= 2 \text{Re}(e^{i\psi_2} \varphi_2) \epsilon + \left(-\text{Re}(g_{0021}(0))H_{000001} - \text{Re}(g_{1011}(0))H_{000010} \right. \\ &\quad \left. + H_{001100} + \text{Re}(e^{2i\psi_2} H_{002000}) \right) \epsilon^2, \quad \psi_2 \in [0, 2\pi]. \end{aligned}$$

6.4 Transcritical-Hopf bifurcation

Compared with the fold-Hopf bifurcation in Section 6.2, the eigenvalues, eigenfunctions, the homological equation, and the functions \mathcal{H} , K and R remain unchanged. It is only the ODE on the center manifold \dot{z} that changes to the normal form (75). Furthermore, also the critical normal form coefficients for the transcritical-Hopf bifurcation remain the same as for the fold-Hopf bifurcation. Therefore, we proceed only with the parameter-related equations.

Collecting the coefficients of the $z_0\beta$ and $z_1\beta$ terms in the homological equation we obtain the systems

$$\begin{aligned} -A^{\odot\star} j H_{10010} &= A_1(\varphi_0, K_{10}) r^{\odot\star} - j \varphi_0, \\ -A^{\odot\star} j H_{10001} &= A_1(\varphi_0, K_{01}) r^{\odot\star}, \\ (i\omega_0 - A^{\odot\star}) j H_{01010} &= A_1(\varphi_1, K_{10}) - i\omega_1 j \varphi_1 r^{\odot\star}, \\ (i\omega_0 - A^{\odot\star}) j H_{01001} &= A_1(\varphi_1, K_{01}) - (1 + i\omega_2) j \varphi_1 r^{\odot\star}. \end{aligned} \tag{96}$$

Let

$$K_\mu = \gamma_{1\mu} e_1 + \gamma_{2\mu} e_2, \quad \mu = (10), (01), \tag{97}$$

where $e_1 = (1, 0)$, $e_2 = (0, 1)$ and $\gamma_{i\mu} (i = 1, 2) \in \mathbb{R}$. To determine $\gamma_{i\mu} (i = 1, 2)$ we substitute (97) into (96). Then by (FSC) we obtain the system

$$\begin{pmatrix} p_0 A_1(\varphi_0, e_1) & p_0 A_1(\varphi_0, e_2) \\ \text{Re}(p_1 A_1(\varphi_1, e_1)) & \text{Re}(p_1 A_1(\varphi_1, e_2)) \end{pmatrix} \begin{pmatrix} \gamma_{110} & \gamma_{210} \\ \gamma_{101} & \gamma_{201} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In order to make the last two systems in (96) consistent we must have that

$$\begin{aligned} \omega_1 &= \text{Im}(p_1 A_1(\varphi_1, K_{10})), \\ \omega_2 &= \text{Im}(p_1 A_1(\varphi_1, K_{01})). \end{aligned}$$

6.4.1 Neimark-Sacker predictors

The predictors for the Hopf and transcritical bifurcation curves, as well as those for the Neimark-Sacker bifurcation curves (including the cycle periods), can be easily obtained using the asymptotics from [Sections 5.4.1](#) and [5.4.2](#). In particular, to approximate the periodic orbits along the Neimark-Sacker curves, we substitute $z_1 = \epsilon e^{i\psi}$ and [\(77\)](#) into [\(86\)](#). This gives the following linear approximations:

$$u = \left(\mp \sqrt{\frac{g_{011}(0)}{g_{200}(0)}} \varphi_0 + 2 \operatorname{Re}(e^{i\psi} \varphi_1) \right) \epsilon \quad \psi \in [0, 2\pi].$$

7 Computation of derivatives for discrete DDEs

All predictors described in the previous sections are implemented in version 3.2a of `DDE-BifTool` for models of the type [\(1\)](#). The discrete DDE [\(1\)](#) is a particular instance of [\(28\)](#) with $h = \tau_m$ and

$$F(\varphi, \alpha) = f(\Xi\varphi, \alpha),$$

where the linear evaluation operator $\Xi : X \rightarrow \mathbb{R}^{n \times (m+1)}$ is defined by

$$\Xi\varphi := (\varphi(-\tau_0), \varphi(-\tau_1), \dots, \varphi(-\tau_m)). \quad (98)$$

with the convention $\tau_0 := 0$. In particular, by the chain rule,

$$D_1 F(0, 0)\varphi = D_1 f(0, 0)\Xi\varphi = \sum_{j=0}^m D_{1,j} f(0, 0)\varphi(-\tau_j), \quad \varphi \in X,$$

with $M_j := D_{1,j} f(0, 0) \in \mathbb{R}^{n \times n}$ the partial derivative of f at the origin with respect to its j th state argument. So, if [\(1\)](#) has an equilibrium at the origin for $\alpha = 0$, then the linear part of the splitting [\(11\)](#) at $\alpha = 0$ is precisely the right-hand side of the above equation. Therefore $\zeta : [0, h] \rightarrow \mathbb{R}^{n \times n}$ must be such that

$$\langle \zeta, \varphi \rangle = \sum_{j=0}^m M_j \varphi(-\tau_j), \quad \forall \varphi \in X.$$

Hence ζ has jump discontinuities M_j at the points τ_j for $j = 0, \dots, m$ and is constant otherwise. So, in this case the characteristic matrix [\(20\)](#) is given by

$$\Delta(z) = zI - \sum_{j=0}^m M_j e^{-z\tau_j}, \quad z \in \mathbb{C}.$$

The multilinear forms appearing in [\(78\)](#) can be expressed in terms of the derivatives of the function $f : \mathbb{R}^{n \times (m+1)} \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ from [\(1\)](#). For $r, s \geq 0$ with $r + s \geq 1$ the mixed derivative of order $r + s$ of f at $(0, 0)$ is an $(r + s)$ -linear form on $[\mathbb{R}^{n \times (m+1)}]^r \times [\mathbb{R}^p]^s$, with the understanding that at most one factor may be absent in case $r = 0$ or $s = 0$. Let Q, Q^1, \dots, Q^r be matrices in $\mathbb{R}^{n \times (m+1)}$ and let $\alpha, \alpha^1, \dots, \alpha^s$ be vectors in \mathbb{R}^p . Then this derivative acts as

$$D_1^r D_2^s f(0, 0)(Q^1, \dots, Q^r, \alpha^1, \dots, \alpha^s) = \sum_{j, k, \ell} \frac{\partial^{r+s} f(Q, \alpha)}{\partial q_{j_1 k_1} \dots \partial q_{j_r k_r} \partial \alpha_{\ell_1} \dots \partial \alpha_{\ell_s}} \Big|_{(Q, \alpha) = (0, 0)} q_{j_1 k_1}^1 \dots q_{j_r k_r}^r \alpha_{\ell_1}^1 \dots \alpha_{\ell_s}^s, \quad (99)$$

where the multidimensional sum runs over

$$1 \leq j_1, \dots, j_r \leq n, \quad 0 \leq k_1, \dots, k_r \leq m, \quad 1 \leq \ell_1, \dots, \ell_s \leq p.$$

The multilinear forms appearing in (78), as well as (80), are computed from (99) by composition with Ξ from (98) as

$$D_1^r D_2^s F(0, 0)(\varphi_1, \dots, \varphi_r, \alpha_1, \dots, \alpha_s) = D_1^r D_2^s f(0, 0)(\Xi\varphi_1, \dots, \Xi\varphi_r, \alpha_1, \dots, \alpha_s),$$

for $\varphi_1, \dots, \varphi_r \in X$ and $\alpha_1, \dots, \alpha_s \in \mathbb{R}^p$. For given r and s the multidimensional array of partial derivatives inside the sum in (99) is of course symmetric under permutation of the state indices $j_1 k_1, \dots, j_r k_r$ and the parameter indices ℓ_1, \dots, ℓ_s . This can be exploited for efficient storage and access.

8 Examples

In this section we will demonstrate the correctness of the normal form coefficients and the accuracy of the predictors in four different models. We do this twofold. Firstly, by comparing the predictors in parameter-space with the computed in DDE-BifTool bifurcation curves, and, secondly, by performing simulations near the bifurcation point under consideration. The simulation is done either with the build-in routine `dde23` of MATLAB or with the Python package `pydelay` [24]. The latter gives significant speed performance when considering simulation over longer time intervals. This usually is the case when one wants to demonstrate the existence of stable invariant manifolds. Since in this section only the main results are given, we provide details (including simulation results) in the [Supplement](#). Furthermore, the source code of the examples has been included into the DDE-BifTool software package. This will hopefully provide a good starting point when considering other models.

8.1 Generalized Hopf bifurcation in a coupled FHN neural system with delay

In [65] the following system is considered

$$\begin{cases} \dot{u}_1(t) = -\frac{u_1^3(t)}{3} + (c + \alpha)u_1^2(t) + du_1(t) - u_2(t) + 2\beta f(u_1(t - \tau)), \\ \dot{u}_2(t) = \varepsilon(u_1(t) - bu_2(t)). \end{cases} \quad (100)$$

Here (u_1, u_2) is the completely synchronous solution of the three coupled FitzHugh–Nagumo (FHN) neuron system

$$\begin{cases} \dot{u}_1(t) = -\frac{u_1^3(t)}{3} + (c + \alpha)u_1^2(t) + du_1(t) - u_2(t) + \beta [f(u_3(t - \tau)) + f(u_5(t - \tau))], \\ \dot{u}_2(t) = \varepsilon(u_1(t) - bu_2(t)), \\ \dot{u}_3(t) = -\frac{u_3^3(t)}{3} + (c + \alpha)u_3^2(t) + du_3(t) - u_4(t) + \beta [f(u_3(t - \tau)) + f(u_5(t - \tau))], \\ \dot{u}_4(t) = \varepsilon(u_3(t) - bu_4(t)), \\ \dot{u}_5(t) = -\frac{u_5^3(t)}{3} + (c + \alpha)u_5^2(t) + du_5(t) - u_6(t) + \beta [f(u_1(t - \tau)) + f(u_3(t - \tau))], \\ \dot{u}_6(t) = \varepsilon(u_5(t) - bu_6(t)), \end{cases} \quad (101)$$

where α, β measure the synaptic strength in self-connection and neighborhood-interaction, respectively. The parameters b and ε are assumed to be positive such that $0 < b < 1$ and $0 < \varepsilon \ll 1$. The function f is a sufficiently smooth sigmoid amplification function and $\tau > 0$ represents the time delay in signal transmission. For the derivation of (100) from the system (101), as well as for stability conditions of the completely synchronous solution, we refer to [65]. In that article a generalized Hopf point was analyzed using the traditional formal adjoint method and the two-step center manifold reduction,

see [31]. Numerical simulations were made to confirm their results. For this (β, α) are taken as the unfolding parameters and the parameters

$$b = 0.9, \quad \varepsilon = 0.08, \quad c = 2.0528, \quad d = -3.2135, \quad \tau = 1.7722$$

are fixed. The sigmoid amplification function $f(u) = \tanh(u)$ is used.

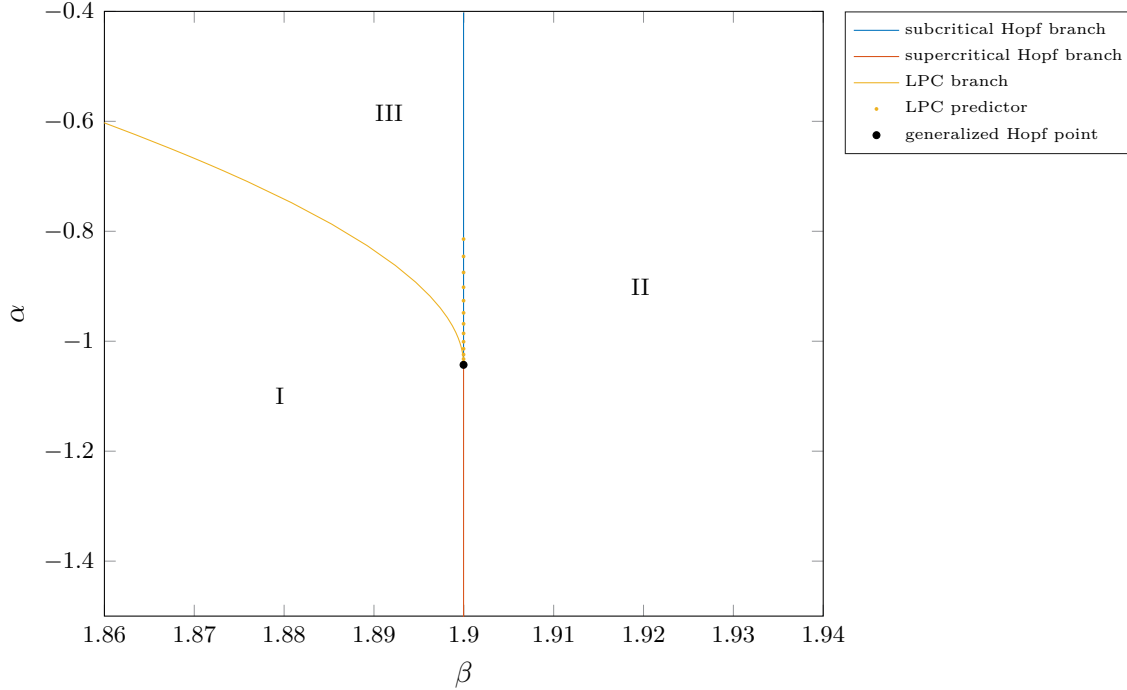


Figure 1: Bifurcation diagram near the generalized Hopf point in the system (100) with unfolding parameters (β, α) . The bifurcation curves are nearly identical to those in the bifurcation diagram of the topological normal form as presented in [43, page 314].

According to [65], a generalized Hopf point is present at the origin with the parameter values $(\beta, \alpha) = (1.9, -0.9710)$. We took this point and calculated its stability and the corresponding normal form coefficients. Although we do confirm that the point under consideration is a Hopf point, the first Lyapunov coefficient does not vanish and we conclude that the point cannot be a generalized Hopf point. However, the simulation in [65] do suggest a generalized Hopf point for nearby parameter values. Therefore we continued the Hopf point in (β, α) . Then a generalized Hopf point is located at $(\beta, \alpha) = (1.9, -1.0429)$ with *negative* second Lyapunov coefficient $\ell_2(0) = -15.6733$, indicating the existence of a stable steady state inside a unstable cycle, which in turn is located inside a stable cycle. We remark that the second Lyapunov coefficient found in [65] is *positive*. This contradicts the simulation of the dynamics made in the same article. Indeed when the second Lyapunov coefficient is positive a time-reversal must be taking into account when considering the bifurcation diagram in the case the second Lyapunov coefficient is negative, see [43]. Then the situation of a stable steady state inside a stable cycle (separated by an unstable cycle) does not occur.

Using the predictors from Section 5.1 the fold and LPC bifurcation curves emanating from the generalized Hopf point. In Figure 1 the resulting bifurcation diagram is shown.

8.2 Fold-Hopf bifurcation of the Rose–Hindmarsh model with time delay

In [48] a Rose-Hindmarsh model [33, 34] with time delay in the self-feedback process, which takes the form

$$\begin{cases} \dot{x}(t) = y(t) - ax^3(t) + bx^2(t - \tau) - cz(t) + I_{app}, \\ \dot{y}(t) = c - dx^2(t) - y(t), \\ \dot{z}(t) = r(S(x(t) - \chi) - z(t)), \end{cases} \quad (102)$$

is considered. Here x represents membrane potential, y represents a recovery variable, z denotes the adaption current, and $a, b, c, d > 0, S$ and χ are real constants. The external current I_{app} and r are control parameters, and τ denotes the synaptic transmission delay. The constants a, b, c, d, χ and r are fixed. Let (x_*, y_*, z_*) be a steady state of (102), then

$$y_* = c - dx_*^2, \quad z_* = S(x_* - \chi). \quad (103)$$

The conditions for a fold-Hopf bifurcation have been derived in [48] analytically. Indeed, let S be arbitrary and set

$$\begin{aligned} x_* &= \frac{1}{3a} \left(b - d \pm \sqrt{(b - d)^2 - 3acS} \right), \\ I_{app} &= x_*^2(ax_* - b + d) + c(S(x_* - \chi) - 1), \\ A &= x_*^2 \left((3ax_* + 2d)^2 - 4b^2 \right) - 2rx_*(2dx_* - 1)(3ax_* - 2b + 2d) \\ &\quad + r^2(4dx_*(-2bx_* + dx_* - 1) + 1), \\ B &= 9a^2x_*^4 + 2rx_*(3ax_* - 2b + 2d) - 4b^2x_*^2 - 4dx_* + r^2 + 1, \\ \omega_{1,2} &= \sqrt{-B \pm \sqrt{B^2 - 4A}}. \end{aligned} \quad (104)$$

Then a fold-Hopf bifurcation occurs when

$$\tau = \begin{cases} \frac{1}{\omega_{1,2}} (\arcsin Y + 2k\pi), & Z \geq 0, \\ \frac{1}{\omega_{1,2}} (\pi - \arcsin Y + 2k\pi), & Z \leq 0, \end{cases}$$

where $k = 0, 1, 2, \dots$ and

$$\begin{aligned} Y &= \frac{\omega_{1,2}}{2b} \left(\frac{r(2b - 2d - 3ax_*)}{r^2 + \omega_{1,2}^2} + \frac{2d}{\omega_{1,2}^2 + 1} - \frac{1}{x_*} \right), \\ Z &= \frac{\omega_{1,2}}{2b} \left(\frac{r^2(2b - 2d - 3ax_*)}{r^2 + \omega_{1,2}^2} + \frac{2d}{\omega_{1,2}^2 + 1} + 3ax_* \right). \end{aligned}$$

In [48] the parameters values

$$a = 1.0, \quad b = 3.0, \quad c = 1.0, \quad d = 5.0, \quad \chi = -1.6, \quad r = 0.001 \quad (105)$$

are fixed. It follows that a fold-Hopf bifurcation is located at

$$x_* = 0.1308, \quad S = -0.57452592, \quad \tau = 5.768830916,$$

and $I_{app}, (y_*, z_*)$ given by (104) and (103), respectively. To unfold the singularity the ‘parameters’ (x_*, S) are used, see [48]. Here we will take the more natural unfolding parameter (I_{app}, S) . Calculating the stability with DDE-BifTool gives the eigenvalues

$$0.001 \pm 1.0081i, \quad -0.000 + 0.000i.$$

All other eigenvalues lie in the open left half of the complex plane. Calculating the normal form coefficients reveals that

$$s = \text{sgn}(g_{200}(0)g_{011}(0)) = \text{sgn}(1.8487\text{e}-05), \quad \theta(0) = \frac{\text{Re}(g_{110}(0))}{g_{200}(0)} = -139.0315$$

and

$$e(0) = \text{Re} \left[g_{210}(0) + g_{110}(0) \left(\frac{\text{Re} g_{021}(0)}{g_{011}(0)} - \frac{3}{2} \frac{g_{300}(0)}{g_{200}(0)} + \frac{g_{111}(0)}{2g_{011}(0)} \right) - \frac{g_{021}(0)g_{200}(0)}{g_{011}} \right] = 15.6941. \quad (106)$$

Since $s = 1$ and $\theta(0) < 0$, a global bifurcation curve or invariant tori are present for parameters sufficiently close to the bifurcation, see [43, page 342]. However, since the sign of $e(0)$ is *positive* the tori are unstable. Thus according to our analysis the simulated torus in [48] cannot be attributed to the fold-Hopf bifurcation.

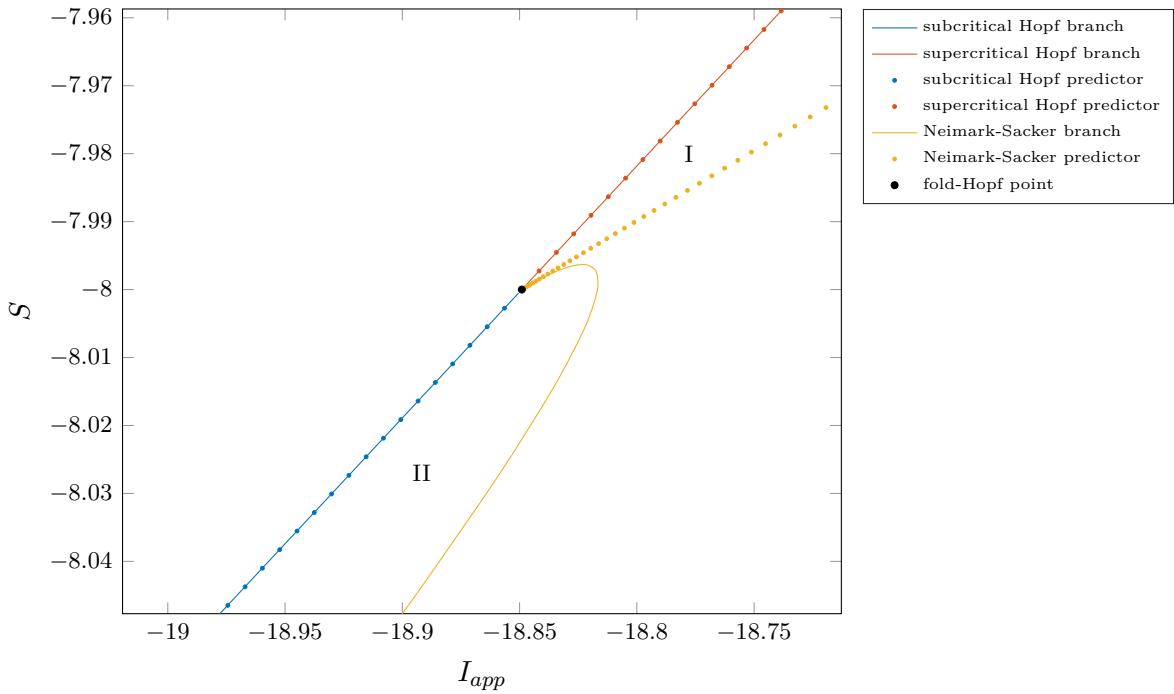


Figure 2: Bifurcation diagram near the fold-Hopf point in (102) with $(r, S) = (1.4, -8)$. The fold branch is not included here since it is indistinguishable from the Hopf curve at this scale.

For demonstration purposes, we take the parameters $r = 1.4$ and $S = -8$, while keeping the other parameters as in (105). Then a fold-Hopf bifurcation is located at $x_\star = 1.0972$, $\tau = 0.9402$, I_{app} as in (104), and (y_\star, z_\star) given by (103). The leading eigenvalues become

$$0.000 \pm 5.6042i, \quad 0.000 + 0.000i,$$

while the normal form coefficients are given by

$$s = \text{sgn}(1.7700), \quad \theta(0) = -0.1569 \quad \text{and} \quad e(0) = -0.0378.$$

Thus the sign of s and $\theta(0)$ remain unchanged. However, since the sign of $e(0)$ is *negative*, there is a time reversal to take into account. Therefore, we expect a stable torus to be present for nearby parameter values. Using the predictors from Section 5.2, we successfully continued the fold, Hopf, and Neimark-Sacker bifurcation curves emanating from the point, see Figure 2.

8.3 Hopf-Hopf and generalized Hopf bifurcations in Active control system

Active control system is used to control the response of structures to internal or external excitation. The mathematical model with time delay can be described as follows [50]

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) + ux(t - \tau) + v\dot{x}(t - \tau) = \tilde{f}(t). \quad (107)$$

Here $x(t)$ is the displacement of the controlled system, $m > 0$ is the mass, c and k are the damping and the stiffness, respectively, τ is the time delay represented in the relative displacement feedback loop and in the relative velocity feedback loop, u and v are feedback strengths, respectively, and \tilde{f} represents the external excitation. Let $t^* = \sqrt{k/mt}$, $\zeta = c/2m\sqrt{m/k}$, $g_u = u/k$, $g_v = v/m\sqrt{m/k}$ and $f(t) = \tilde{f}(t)/k$. Then equation (107) becomes

$$\ddot{x}(t) + 2\zeta\dot{x}(t) + x(t) + g_u x(t - \tau) + g_v \dot{x}(t - \tau) = f(t),$$

where the asterisks are omitted for simplicity. Following [16] and [50] we consider the case when f is replaced by a nonlinear position time delay feedback given by $\beta x^3(t - \tau)$, see also [63]. As in [16] we fix the parameters

$$g_u = 0.1, \quad g_v = 0.52, \quad \beta = 0.1$$

and take ζ and τ as control parameters. Let $\dot{x}(t) = y(t)$, then we obtain

$$\begin{cases} \dot{x}(t) = \tau y(t), \\ \dot{y}(t) = \tau (-x(t) - g_u x(t - 1) - 2\zeta y(t) - g_v y(t - 1) + \beta x^3(t - 1)). \end{cases} \quad (108)$$

Here the delay is scaled by using the transformation of time $t \rightarrow t/\tau$. In this way the delay can be treated as an ordinary parameter.

The trivial equilibrium undergoes a Hopf-Hopf bifurcation at the parameter values

$$(\zeta_c, \tau_c) = (-0.016225, 5.89802), \quad (109)$$

see [16] for the derivation. Using `DDE-BifTool` we manually construct the Hopf-Hopf point and compute its stability and normal form coefficients. We obtain the eigenvalues $0.0000 \pm 4.5275i$ and $-0.0000 \pm 7.6449i$. The quadratic critical normal form coefficients are

$$\begin{aligned} g_{2100}(0) &= -0.0915 + 0.1214i, & g_{1110}(0) &= 0.2151 + 0.3876i, \\ g_{1011}(0) &= -0.3084 + 0.4096i, & g_{0021}(0) &= 0.1813 + 0.3268i. \end{aligned}$$

From

$$(\operatorname{Re} g_{2100}(0))(\operatorname{Re} g_{0021}(0)) = -0.0166 < 0,$$

we conclude that this Hopf-Hopf bifurcation is of ‘difficult’ type, see [43]. Furthermore, since the quantities

$$\theta = \theta(0) = \frac{\operatorname{Re} g_{1011}(0)}{\operatorname{Re} g_{0021}(0)} = -1.7009, \quad \delta = \delta(0) = \frac{\operatorname{Re} g_{1101}(0)}{\operatorname{Re} g_{2100}(0)} = -2.3517$$

are such that $\theta < 0$, $\delta < 0$, $\theta\delta > 1$ it follows that we are in case VI. We continue the Neimark-Sacker and Hopf bifurcation curves emanating from the Hopf-Hopf point using the predictors from Section 5.3. In Figure 3 a close-up is given near the Hopf-Hopf point comparing the computed curves with the predictors in parameter space.

Using the detection capabilities of `DDE-BifTool` one additional Hopf-Hopf point and three generalized Hopf points are located on the continued Hopf branches. The normal form coefficients of the second Hopf-Hopf point are such that

$$(\operatorname{Re} g_{2100}(0))(\operatorname{Re} g_{0021}(0)) = 1.7331e-04 > 0$$

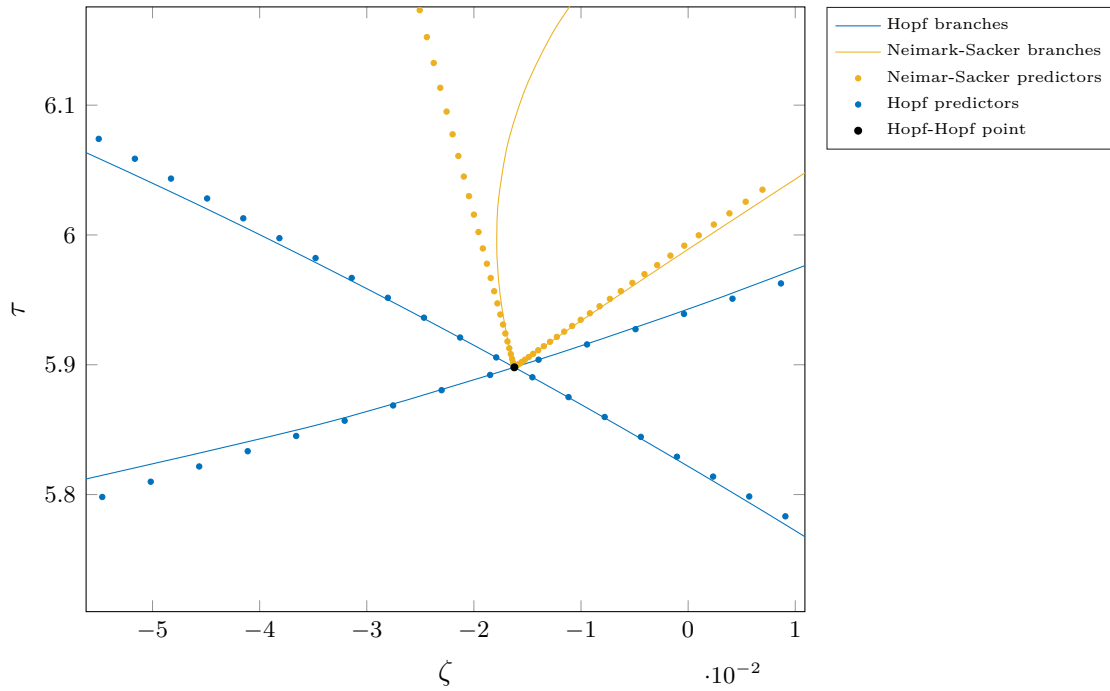


Figure 3: Bifurcation diagram near the Hopf-Hopf point at parameter values (109) in an active control system with time delay given by (108). There are two supercritical Hopf curves (blue) and two Neimark-Sacker curves (yellow). We see that the predictors (dotted) give good approximations near the codimension two point.

and

$$\theta \geq \delta > 0, \quad \theta\delta > 1.$$

We conclude that we are in case I of the ‘simple’ type, see [43, page 360]. Therefore, no stable invariant two-dimensional torus is predicted for nearby parameter values, only two stable period orbits expected. Using the predictors from Sections 5.1 and 5.3 we can easily continue the codimension one cycle bifurcations from the located degenerate Hopf points, showing complicated bifurcation diagram in Figure 4.

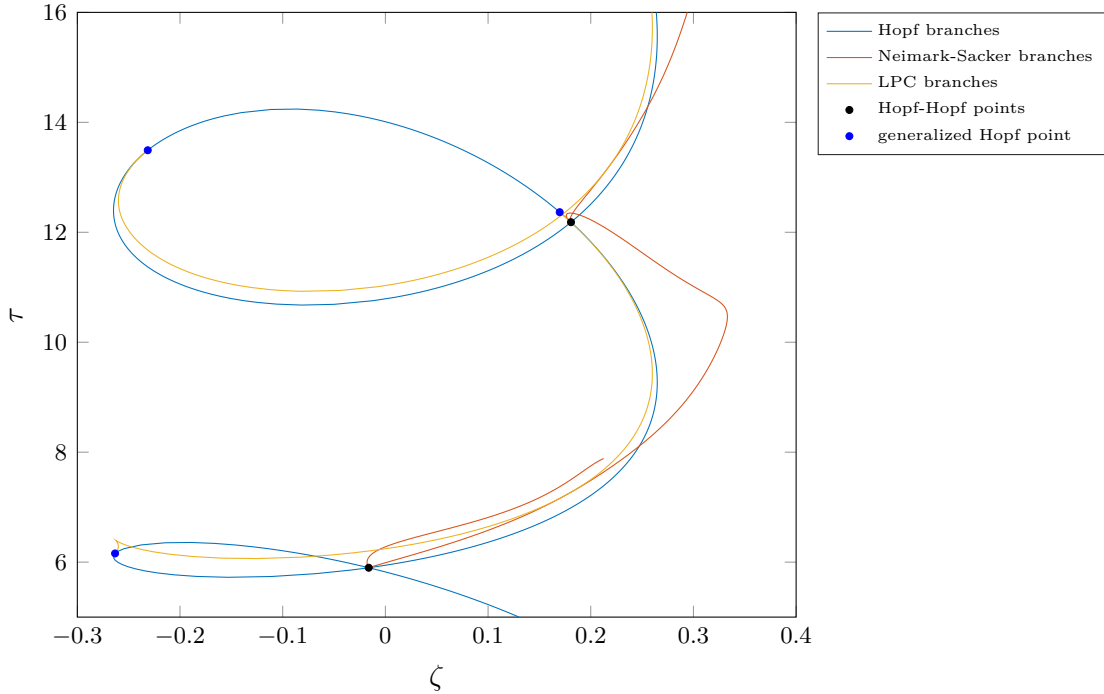


Figure 4: Bifurcation diagram obtained by continuing Hopf-Hopf, Neimark-Sacker and LPC bifurcation curves in the active control system (108) using the predictors from Sections 5.1 and 5.3 combined with the continuation capabilities from DDE-BifTool. Two Hopf-Hopf points are connected by a Neimark-Sacker bifurcation curve. Also two of the three generalized Hopf points are connected by a single LPC curve.

8.4 Transcritical-Hopf bifurcation in Van der Pol’s oscillator with delayed position and velocity feedback

In [3] a generalization of Van der Pol’s oscillator with delayed feedback

$$\ddot{x}(t) + \varepsilon(x^2(t) - 1)\dot{x}(t) + x(t) = g(\dot{x}(t - \tau), x(t - \tau)), \quad 0 < \tau < \infty, \quad (110)$$

is considered. Here $g \in C^3$ satisfies the conditions $g(0, 0) = 0$, $g_{\dot{x}}(0, 0) = a$ and $g_x(0, 0) = b$. The linearization of equation (110) around the trivial solution $x = 0$ gives

$$\ddot{x}(t) - \varepsilon\dot{x}(t) + x(t) = a\dot{x}(t - \tau) + bx(t - \tau).$$

From which we obtain the characteristic equation

$$\Delta(\lambda, \tau) = \lambda^2 - \varepsilon\lambda + 1 - (a\lambda + b)e^{-\lambda\tau} = 0.$$

Let

$$b = 1, \quad \tau = \tau_0 \neq \varepsilon + a, \quad \varepsilon^2 - a^2 < 2, \quad (111)$$

then the characteristic equation has a simple zero and a pair of purely imaginary roots $\lambda = \pm i\omega_0$. Here ω_0 and τ_0 are defined by

$$\omega_0 = \sqrt{2 - \varepsilon^2 + a^2}, \quad \tau_0 = \frac{1}{\omega_0} \arccos\left(\frac{1 - (1 + \varepsilon a)\omega_0^2}{a^2\omega_0^2 + 1}\right),$$

see [3, Proposition 2.1]. We set the function g to

$$g(\dot{x}(t - \tau), x(t - \tau)) = (1 + \mu_1)x(t - \tau) - 0.2\dot{x}(t - \tau) - 0.2x(t - \tau)^2 \\ - 0.2x(t - \tau)\dot{x}(t - \tau) - 0.2x(t - \tau)^2 + 0.5x(t - \tau)^3$$

and $\varepsilon = 0.3$. Then the conditions (111) are satisfied and

$$\omega_0 \approx 1.396424004376894, \quad \tau_0 \approx 1.757290761249588. \quad (112)$$

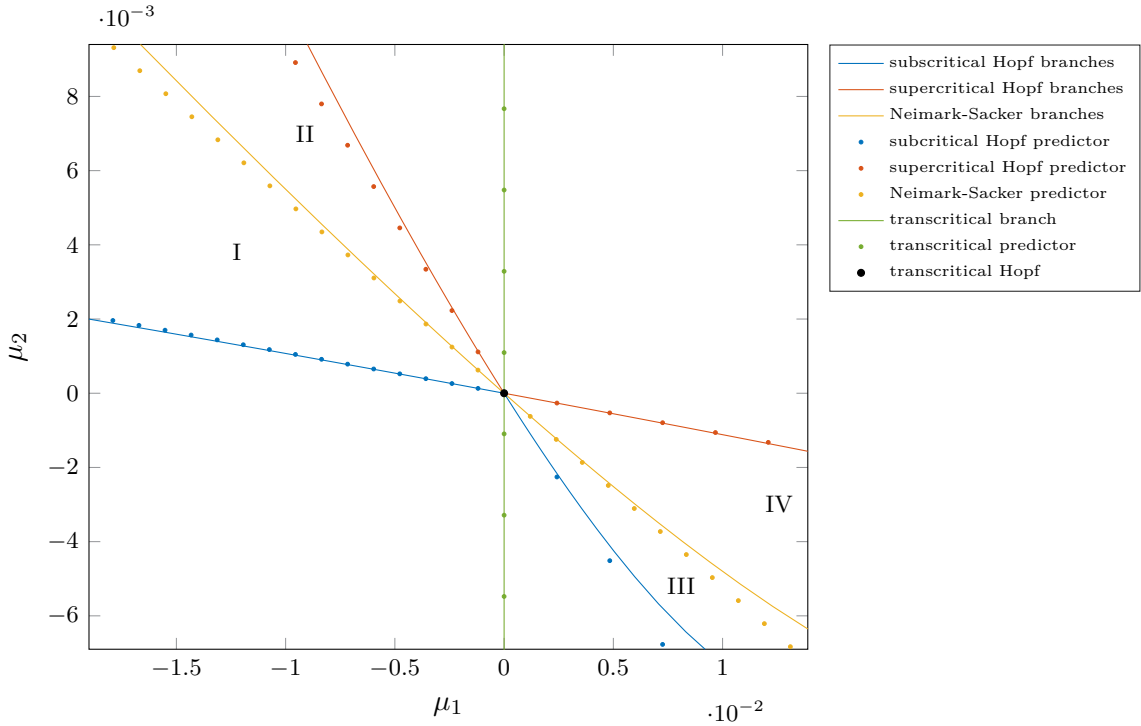


Figure 5: Bifurcation diagram near the transcritical-Hopf bifurcation in the delayed Van der Pol's oscillator given by (113). There are two supercritical Hopf curves (blue), two subcritical Hopf curves (red), two Neimark-Sacker curves (yellow) and one transcritical curve (green). We see that the predictors (dotted) give good approximation for nearby values.

To analyze the system with `DDE-BifTool` we set $y(t) = \dot{x}(t)$ and transform the time with $t \rightarrow t/\tau$ to obtain the two-component system

$$\begin{cases} \dot{x}(t) = (\tau_0 + \mu_2) y(t), \\ \dot{y}(t) = (\tau_0 + \mu_2) [-x(t) - \varepsilon(x^2(t) - 1)y(t) + (1 + \mu_1)x(t - 1) - 0.2y(t - 1) \\ - 0.2x^2(t - 1) - 0.2x(t - 1)y(t - 1) - 0.2y^2(t - 1) + 0.5x^3(t - 1)]. \end{cases} \quad (113)$$

Here we introduced the unfolding parameters $(\mu_1, \mu_2) := (b - 1, \tau - \tau_0)$ to translate the singularity to the origin. One immediately sees that the trivial equilibrium $(\dot{x}, x) = (0, 0)$ is an equilibrium for all parameter values (μ_1, μ_2) . Therefore, the parameter-dependent normal form for the generic fold-Hopf cannot be used here. Instead the normal form for the transcritical-Hopf bifurcation must be used. Using `DDE-BifTool` we compute the stability and the normal form coefficients. The leading eigenvalues are $0.000 + 0.000i$ and $-0.000 + 2.4539i$, where $2.4539 \approx \omega_0 \tau_0$, see (112). Furthermore, the normal form coefficients are such that

$$g_{011}(0) \times \operatorname{Re}(g_{110}(0)) = 0.4241 \times \operatorname{Re}(-0.1337 + 0.2672i) < 0.$$

Therefore, there are two Neimark-Sacker bifurcation curves predicted, see Section 5.4. Using the predictors from Section 5.4 we continue the transcritical, Hopf and Neimark-Sacker bifurcation curves emanating from the transcritical-Hopf bifurcation point. In Figure 5 the bifurcation diagram is shown.

9 Concluding remarks

We have provided explicit formulas for the normal form coefficients needed to initialize codimension one equilibrium and nonhyperbolic cycle bifurcations emanating from generalized Hopf, fold-Hopf, Hopf-Hopf and transcritical-Hopf points in DDEs. Applications to four different models were given, confirming the correctness of the derivation of the normal form coefficients and the asymptotics. A paper providing a second-order predictor for the homoclinic orbits emanating from the generic and transcritical codimension two Bogdanov-Takens bifurcations in DDEs, along the lines of [45], is in preparation.

Our proof of the existence of a smooth parameter-dependent center manifold is given in the general context of perturbation theory for dual semigroups (sun-star calculus). Consequently the applicability of this result extends beyond classical DDEs, although here we did restrict to the case of an eventually compact \mathcal{C}_0 -semigroup on a sun-reflexive state space. It follows that the results from Sections 3.2 to 3.5 are valid as well for other classes of delay equations such as renewal equations (also known as Volterra functional equations) and systems of mixed type [11].

Furthermore, in [15, 59] the technique was used to calculate the critical normal form coefficients for Hopf and Hopf-Hopf bifurcations occurring in neural field models with propagation delays. For these models sun-reflexivity is lost, which is typical for delay equations in abstract spaces or with infinite delay. However, it is often possible to overcome this functional analytic complication, so dual perturbation theory can still be employed successfully [12, 13, 38, 59]. It has also been used in the context of semilinear hyperbolic systems [47].

It is demonstrated - at a formal level - in [55] that the normalization technique described in Section 4 still works for DDEs with state-dependent delays. However, as already mentioned in Remark 21, for DDEs the nonlinearity generally does *not* depend differentiably on the delay parameters. Therefore, in the case of state-dependent DDEs it is generally not possible to justify differentiation of the nonlinearity with respect to the state, let alone to rely on higher order smoothness. So, as far as we know there is still no proof of the validity of the normalization technique for state-dependent DDEs.

Returning to the setting of classical DDEs, the most obvious next challenge is to derive normal forms for bifurcations of periodic orbits by generalizing [9, 10, 44] to DDEs. The resulting formulas can then be implemented in `DDE-BifTool` to facilitate numerical bifurcation analysis of periodic orbits in classical DDEs.

Acknowledgments

The authors would like to thank Prof. Odo Diekmann (Utrecht University) for very useful discussions on parameter-dependent perturbation of linear semigroups. We also thank Prof. Peter De Maesschalck (Hasselt University) for supporting this research project.

SUPPLEMENTARY MATERIALS FOR:
Switching to nonhyperbolic cycles from codimension two
bifurcations of equilibria of delay differential equations

M.M. Bosschaert* S.G. Janssens† Yu.A. Kuznetsov‡

March 19, 2019

In this supplement we provide walkthroughs of the examples given in [Section 8](#) with `DDE-BifTool`¹ [[21](#)]. These walkthroughs enable other researchers to reproduce the results obtained in the main text.

Additionally, we will show the code used for simulation near the bifurcations points under consideration. Either using the build-in routine `dde23` from `MATLAB` [[53](#)] or the Python package `pydelay`² [[24](#)]. Other DDE models, undergoing one of the degenerate Hopf bifurcations treated in this paper, can easily be studied by making minor modifications to the given code.

The focus will be on the initialization and continuation of the various codimension one equilibrium and cycles bifurcation curves emanating from the degenerate Hopf points and on simulation near the bifurcation points. For a complete overview of the capabilities and functionality for `DDE-BifTool`, we refer to the online tutorials files and also the manual and the references therein.

All code has been included into the `DDE-BifTool` package version 3.2a on the SourceForge repository and can be executed without the need to copy and paste. Note that the code is tested on `MATLAB` 2018b and `GNU Octave` 4.2.2. Different results may occur with other versions of `MATLAB` and `GNU Octave`.

S1 Generalized Hopf bifurcation in a coupled FHN neural system with delay

A completely synchronous solution of the three coupled FitzHugh-Nagumo (FHN) neuron system is given by the system

$$\begin{cases} \dot{u}_1(t) = -\frac{u_1^3(t)}{3} + (c + \alpha)u_1^2(t) + du_1(t) - u_2(t) + 2\beta f(u_1(t - \tau)), \\ \dot{u}_2(t) = \varepsilon(u_1(t) - bu_2(t)). \end{cases} \quad (\text{S1})$$

see [Section 8.1](#) and [[48](#)]. As before, we fix the parameters

$$b = 0.9, \quad \varepsilon = 0.08, \quad c = 2.0528, \quad d = -3.2135, \quad \tau = 1.7722,$$

and take for $f : \mathbb{R} \rightarrow \mathbb{R}$ the sigmoid amplification function $f(u) = \tanh(u)$. The parameters (β, α) are used to unfold the singularity.

*Department of Mathematics, Hasselt University, Diepenbeek Campus, Agoralaan Gebouw D, 3590 Diepenbeek, Belgium (maikel.bosschaert@uhasselt.be).

†Department of Mathematics, Utrecht University, Budapestlaan 6, 3508 TA Utrecht, The Netherlands (s.g.janssens@uu.nl, sj@dydx.nl).

‡Department of Mathematics, Utrecht University, Budapestlaan 6, 3508 TA Utrecht, The Netherlands and Department of Applied Mathematics, University of Twente, Zilverling Building, 7500AE Enschede, The Netherlands (I.A.Kouznetsov@uu.nl).

¹<http://ddebiftool.sourceforge.net/>

²<http://pydelay.sourceforge.net/>

Remark 23. This demonstration can be found in the directory `demos/tutorial/VII/FHN` relative to the main directory of the DDE-BifTool package.

S1.1 Generate system files

Before we start to analyze the system with DDE-BifTool, we first create a *system file*. This file contains the definition of the system (S1), the standard derivatives needed for calculation of the eigenvalues and eigenvectors, the continuation of bifurcation points and cycles, and also the multilinear forms, see Section 7, used for the calculation of the coefficients of the critical and parameter-dependent normal forms. Alternatively, one can only supply the system itself, see Listing S2. Then finite difference is used to approximate the derivatives. However, this is less efficient and accurate, and therefore not recommended. A separate script `gen_sym_FHN.m` is used to create a system file. The most important parts of this script are listed and discussed below.

```

%% Add paths and load sym package if GNU Octave is used
clear
ddebiftoolpath='../..../..../';
addpath(strcat(ddebiftoolpath,'ddebiftool'),...
        strcat(ddebiftoolpath,'ddebiftool_extra_symbolic'));
if dde_isoctave()
    pkg load symbolic
end

%% Create parameter names as strings and define fixed parameters
% The demo has the parameters |beta|, |alpha| and |tau|
parnames={'beta','alpha','tau'};
b=sym(0.9,'r');
epsilon=sym(0.08,'r');
c=sym(2.0528,'r');
d=sym(-3.2135,'r');

%% Create symbols for parameters, states and delays states
% |par| is the array of symbols in the same order as parnames.
% Due to the following two lines we may, for example,
% use either beta or par(1) to refer to the delay.
syms(parnames{:}); % create symbols for beta, alpha and tua
par=cell2sym(parnames); % now beta is par(1) etc

%% Define system using symbolic algebra
% create symbols for u1(t) u1(t-tau), u2(t), u2(t-tau)
syms u1 u1t u2 u2t
du1_dt=-u1^3/3+(c+alpha)*u1^2+d*u1-u2+2*beta*tanh(u1t);
du2_dt=epsilon*(u1-b*u2);

%% Differentiate and generate code (multi-linear forms)
[fstr,derivs]=dde_sym2funcs(...
[du1_dt;du2_dt],... % n x 1 array of derivative symbolic expressions
[u1,u1t;u2,u2t],... % n x (ntau+1) array of symbols for states (current & ...
    delayed)
par,... % 1 x np (or np x 1) array of symbols used for parameters
'filename','sym_FHN_mf',... % optional argument specifying output file
'directional_derivative',false);

%% Differentiate and generate code (directional derivatives)
[fstr,derivs]=dde_sym2funcs(...
[du1_dt;du2_dt],... % n x 1 array of derivative symbolic expressions
[u1,u1t;u2,u2t],... % n x (ntau+1) array of symbols for states (current & ...
    delayed)
par,... % 1 x np (or np x 1) array of symbols used for parameters

```

```
'filename','sym_FHN',... % optional argument specifying output file
'directional_derivative',true);
```

The variable `ddebiftoolpath` is directed to the DDE-BifTool main folder, which should have been extracted somewhere on the computer. Here a path relative to the current working directory is used. Note that although we only use the parameters (β, α) as unfolding parameters, in the current version of DDE-BifTool, we also need to include the delay(s) in the list of parameters. After running the script, the function `dde_sym2funcs` creates two system files `sym_FHN_mf.m` and `sym_FHN.m`. The first file `sym_FHN_mf.m` implements the higher order derivatives as multilinear forms, as explained in [Section 7](#), and therefore the file we will solely be using. The second file `sym_FHN.m` uses directional derivatives to implement the higher order derivatives. The directional derivatives approach *formally* allows the use of state-dependent delays, see [\[55\]](#). Although both approaches yields (up to rounding errors) identical normal form coefficients, multilinear forms are much faster.

S1.2 Loading the DDE-BifTool package

Now that a system file is created we continue with DDE-BifTool to analyze (S1). The code in the following sections highlights the import parts of the file `FHN.m`. DDE-BifTool consists of a set of MATLAB routines. Thus, to start using DDE-BifTool, we only need to add DDE-BifTool directories to the search path.

Listing S1: Add DDE-BifTool scripts to the search path

```
%% Clean workspace and add DDE-BifTool scripts to
% the MATLAB search path
clear; % clear variables
close all; % close figures
ddebiftoolpath='../../../../../../';
addpath(strcat(ddebiftoolpath, 'ddebiftool'),...
        strcat(ddebiftoolpath, 'ddebiftool_extra_psol'),...
        strcat(ddebiftoolpath, 'ddebiftool_extra_nmfm'),...
        strcat(ddebiftoolpath, 'ddebiftool_utilities'));
```

There are four subdirectories added to the search path:

ddebiftool Containing the core files of DDE-BifTool.

ddebiftool_extra_psol An extension for enabling continuation of periodic orbit bifurcations for delay-differential equations with constant or state-dependent delay.

ddebiftool_extra_nmfm An extension for normal form computation.

ddebiftool_utilities Containing various utilities.

S1.3 Set parameter names

The following code allows us to use `ind.beta` instead of remembering the index of the parameter β in the parameter array, and similarly for the other parameters.

```
%% Set parameter names
parnames={'beta','alpha','tau'};
cind=[parnames; num2cell(1:length(parnames))];
ind=struct(cind{:});
```

In this way, fewer mistakes are likely to be made and the code is easier to read.

S1.4 Initialization

Next, we set up the `funcs` structure, containing information about where the system and its derivatives are stored, a function pointing to which parameters are delays, and various other settings.

```
%% Set the funcs structure
% We load the precalculated multilinear forms. These have been
% generated with the file gen_sym_FHN.m.
funcs=set_symfuncs(@sym_FHN_mf, 'sys_tau', @() ind.tau);
```

Alternatively, when no system files have been generated, one could initialize the system (S1) as follows.

Listing S2: Define system without a system file

```
%% Define the system
b=0.9; epsilon=0.08; c=2.0528; d=-3.2135; % fixed parameters
FHN_sys = @(xx,par) [...
-xx(1,1,:).^3/3+(c+par(1,ind.alpha,:)).*xx(1,1,:).^2+d*xx(1,1,:)...
-xx(2,1,:)+2*par(1,ind.beta,:).*tanh(xx(1,2,:));
epsilon*(xx(1,1,:)-b*xx(2,1,:))];
%% Set funcs structure
funcs=set_funcs('sys_rhs',FHN_sys,'sys_tau',@() ind.tau,...
'x_vectorized',true,'p_vectorized',true);
```

Inspecting the output of the `funcs` handle gives.

```
>> funcs

funcs =

  struct with fields:

    sys_rhs: @(x,p)wrap_rhs(x,p,funcs.sys_rhs,funcs.
x_vectorized,funcs.p_vectorized)
    sys_ntau: @()0
    sys_tau: @() ind.tau
    sys_cond: @dummy_cond
    sys_deriv: @(x,p,nx,np,v)dde_gen_deriv(funcs.sys_dirderiv,x,
p,nx,np,v,1)
    sys_dtau: []
    sys_mfderiv: {}
    sys_dirderiv: {[function_handle] [function_handle]}
    sys_dirdtau: []
    x_vectorized: 1
    p_vectorized: 1
    hjac: @(ord)eps^(1/(2+ord))
    sys_unfolding_parameters: []
    tp_del: 0
    sys_deriv_provided: 0
    sys_dirderiv_provided: 0
```

The output shows that no derivative file is supplied. In this case, the derivatives are calculated using finite-difference approximations with the function `dde_dirderiv`. Again, we do not recommend using the latter approach. However, it can be useful for debugging purposes.

S1.5 Stability and normal form coefficients of the generalized-Hopf point

We manually specify a steady-state at the generalized-Hopf point found in [48] and calculate its stability.


```

% construct steady-state point
beta0=1.9; alpha0=-0.9710; tau0=1.7722;
stst=dde_stst_create('x',[0;0]);
stst.parameter(ind.beta) = beta0;
stst.parameter(ind.alpha) = alpha0;
stst.parameter(ind.tau) = tau0;
% Calculate stability
method=df_mthod(funcs,'stst');
stst.stability=p_stabil(funcs,stst,method.stability);

```

Inspecting the `stst.stability` structure yields

```

>> stst.stability.l1(1:6)

ans =

    0.0000 + 0.0720i
    0.0000 - 0.0720i
   -0.0818 + 3.1068i
   -0.0818 - 3.1068i
   -0.3478 + 6.4411i
   -0.3478 - 6.4411i

```

```
>>
```

The eigenvalues confirm that the point under consideration is indeed a Hopf point. Next, we convert the steady-state point to a Hopf point and calculate the normal form coefficients with the function `nmfm_genh`, which implements the coefficients derived in [Section 6.1](#).

```

%% Calculate critical normal form coefficients
hopf=p_tohopf(funcs,stst);
method=df_mthod(funcs,'hopf');
hopf.stability=p_stabil(funcs,hopf,method.stability);
genh=p_togenh(hopf);
genh=nmfm_genh(funcs,genh);

```

The normal form coefficients are stored in the `genh.nmfm` structure.

```

>> genh.nmfm
ans =

    struct with fields:

    L2: -18.1302
    L1: 0.3980

```

```
>>
```

Clearly, the first Lyapunov coefficient (`L1`) is nonzero. It follows that the Hopf point is not degenerate.

S1.6 Continue Hopf point

Since the simulations in [\[48\]](#) do indicate a generalized-Hopf point nearby, we continue the Hopf point.

```

%% Initialize Hopf branch
unfolding_pars=[ind.beta, ind.alpha];
hbr=df_brnch(funcs,unfolding_pars,'hopf');

```

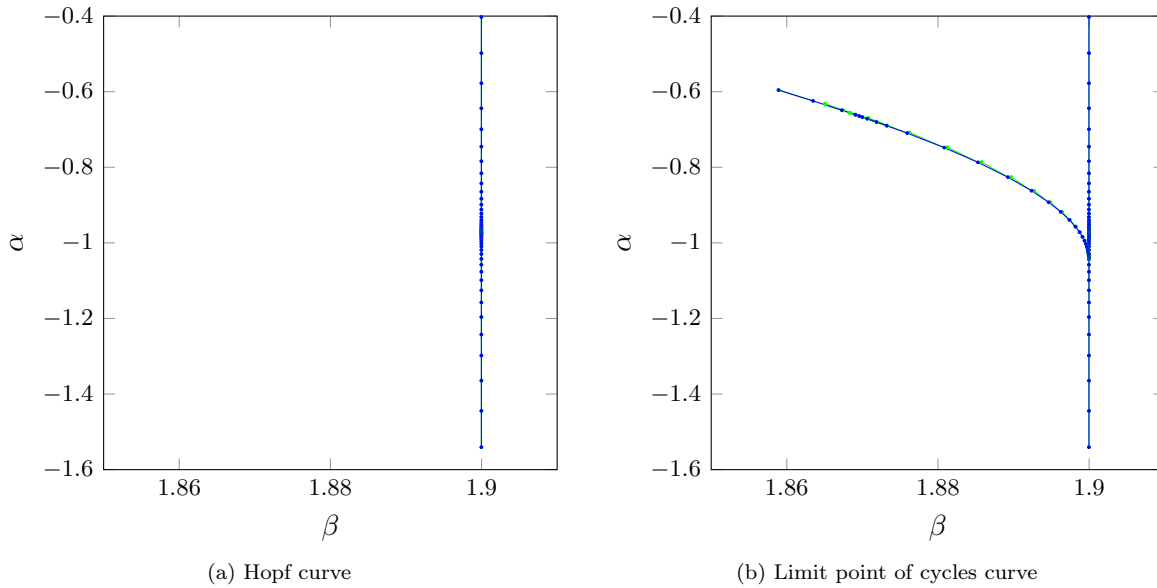


Figure S1: (a) Hopf curve continued from the manually constructed point. (b) LPC curve continued from the detected generalized Hopf point using our predictors.

```

hbr.point=hopf;
hbr.point(2)=hopf;
hbr.point(2).parameter(ind.alpha)=...
hbr.point(2).parameter(ind.alpha)+0.001;
method=df_mthod(funcs,'hopf');
method.point.print_residual_info=1;
hbr.point(2)=p_correc(funcs,hbr.point(2),ind.beta,[],method.point);
%% Continue Hopf branch
figure(1); clf;
hbr=br_contn(funcs,hbr,30);
hbr=br_rvers(hbr);
hbr=br_contn(funcs,hbr,30);
title('Continued Hopf branch');
xlabel('\beta','Interpreter','LaTeX');
ylabel('\alpha','Interpreter','LaTeX');
box on

```

The continued branch `hbr` is shown in [Figure S1a](#).

S1.7 Detect bifurcation points

To detect bifurcation points on the Hopf branch, we use the function `LocateSpecialPoints`.

```

[hbr_wbifs,hopftests,hc2_indices,hc2_types]=...
    LocateSpecialPoints(funcs,hbr);

```

The MATLAB console shows the following output.

```

HopfCodimension2: calculate stability if not yet present
HopfCodimension2: calculate L1 coefficients

```

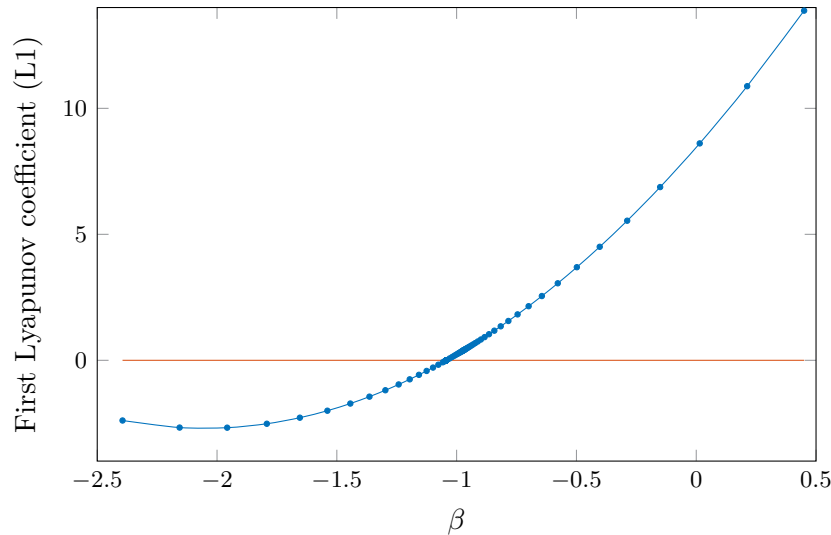


Figure S2: Plot of the test function for a generalized Hopf point.

```
HopfCodimension2: (provisional) 1 gen. Hopf detected.
br_insert: detected 1 of 1: genh. Normalform:
  L2: -15.6733
  L1: -1.6801e-12
```

Thus a generalized Hopf point is indeed present on the Hopf branch `hbr`. The returned branch `hbr_wbifs` contains this point. The array `hc2_indices` is used to subtract the generalized Hopf point below. If there would be more bifurcation points detected, `hc2_types` can be used to inspect their types. Lastly, `hopftests` stores the test functions to detect a bifurcation point. A change in sign in one of these functions indicates a bifurcation. The code below plots the test function for the generalized Hopf point, i.e. the first Lyapunov coefficient (L1), see [Figure S2](#).

```
al=arrayfun(@(x)x.parameter(ind.alpha),hbr_wbifs.point);
figure(2); clf;
plot(al,hopftests.genh(1,:),'.-',al,zeros(size(al)));
xlabel('$\beta$','Interpreter','LaTeX');
ylabel('First Lyapunov coefficient (L1)')
title('Criticality along Hopf bifurcation curve')
```

S1.8 Continue limit point of cycle curve

First, we subtract the detected generalized-Hopf point from the branch.

```
%% Calculate parameter-dependent normal form coefficients
% Select the located generalized Hopf point on the hopf_br_wbifs.
% Then convert the Hopf point to a generalized Hopf point.
% As before, we use the function nmfm_genh to calculate the normal
% form coefficients. By adding the option free_pars and providing
% the unfolding parameters the parameter-dependent normal
% form coefficients are calculated.
hopf=hbr_wbifs.point(hc2_indices); % select generalized Hopf point
genh=p_togenh(hopf);
genh=nmfm_genh(funcs,genh,'free_pars',unfolding_pars);
```

```
genh.nmfm.L1
```

By inspecting the `genh` structure, we obtain the correct parameter values of the generalized Hopf bifurcation.

```
>> genh.parameter

ans =

    1.9000    -1.0429    1.7722

>>
```

To continue the limit point of cycles curve emanating from the generalized Hopf point, we use the function `C1branch_from_C2point`.

```
% Continue LPC curve emanating from generalized-Hopf point
figure(1); [lpcfuncs,lpcbr,~]=C1branch_from_C2point(funcs,genh,...
    unfolding_pars,'codim2',genh.kind,'codim1','P0fold');
nop=50; [lpcbr,suc]=br_contn(lpcfuncs,lpcbr,nop); assert(suc>0)
```

This function uses the file `nmfm_P0fold_from_genh_init.m` in which the predictor, see [Sections 5.1](#) and [6.1.3](#), is implemented. Using this function, a branch with three initial corrected cycles is created which is continued in the standard way, see [Figure S1b](#).

S1.9 Calculate predicted periodic orbits

To compare the computed parameter values and periodic orbits on the branch `lpcbr` with the predictor, we again use the function `C1branch_from_C2point`, but with the additional argument `predictor` set to 1 and `step` to an interval of ε -values. Now the cycles are left uncorrected.

```
% Predictor LPC curve emanating from generalized-Hopf point
[~,lpcbr_pred,~]=C1branch_from_C2point(funcs,genh,...
    unfolding_pars,'codim2',genh.kind,'codim1','P0fold',...
    'step',linspace(0,1,45),'predictor',1);
```

S1.10 Bifurcation diagram

The following code produces the bifurcation diagram presented in the main text, see [Figure 1](#), and has been reproduced here in [Figure S3](#). The figure was exported with the MATLAB and GNU Octave compatible package `matlab2tikz`, see [\[26\]](#).

Listing S3: MATLAB code for bifurcation diagram

```
%% Bifurcation diagram
figure(3); clf; hold on;
% Inline function to subtract parameters
getpars=@(points,ind) arrayfun(@(p)p.parameter(ind),points);
cm=colormap('lines');
L1s=hopftests.genh(1,:); % L1 along the hopf branch
% Plot sub- and supercritical Hopf branches
plot(getpars(hbr_wbifs.point(L1s>0),ind.beta),...
    getpars(hbr_wbifs.point(L1s>0),ind.alpha),'Color',cm(1,:),...
    'DisplayName','subcritical Hopf branch');
plot(getpars(hbr_wbifs.point(L1s<0),ind.beta),...
    getpars(hbr_wbifs.point(L1s<0),ind.alpha),'Color',cm(2,:),...)
```

```

'DisplayName','supercritical Hopf branch');
plot(getpars(lpcbr.point,ind.beta),...
     getpars(lpcbr.point,ind.alpha),'Color',cm(3,:),...
     'DisplayName','LPC branch');
plot(getpars(lpcbr_pred.point,ind.beta),...
     getpars(lpcbr_pred.point,ind.alpha),'.','Color',cm(3,:),...
     'DisplayName','LPC predictor');
plot(getpars(genh,ind.beta),getpars(genh,ind.alpha),'k.',...
     'MarkerSize',8,'DisplayName','generalized Hopf point');
title('Bifurcation diagram near generalized Hopf point')
xlabel('\beta','Interpreter','LaTeX');
ylabel('\alpha','Interpreter','LaTeX');
text(1.8779,-1.1001,'I');
text(1.9130,-0.9008,'II');
text(1.8890,-0.5854,'III');
axis([1.86 1.945 -1.5000 -0.4000])
legend('Location','NorthEast'); box on

```

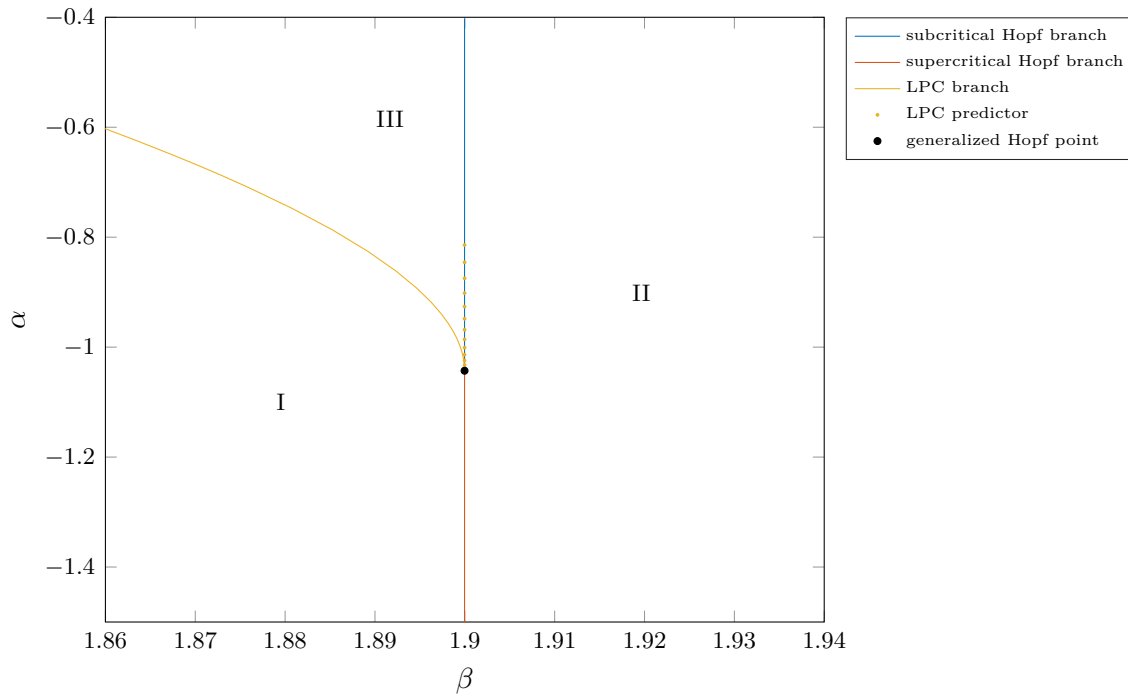


Figure S3: Bifurcation diagram near the generalized Hopf point in the system (S1) with unfolding parameters (β, α) . The bifurcation curves are nearly identical to those in the bifurcation diagram of the topological normal form as presented in [43, page 314].

S1.11 Plot comparing computed and predicted periodic orbits

Lastly, we create a plot to compare the computed and predicted periodic orbits.

```

%% Plot comparing computed and predicted periodic orbits
figure(4); clf; hold on;

```

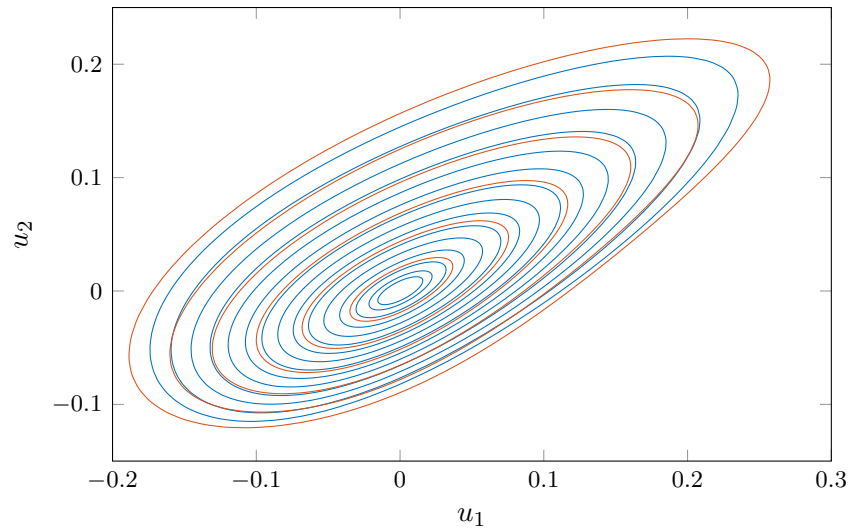


Figure S4: Comparison between computed periodic orbits (blue) and predicted periodic orbits (red) emanating from the generalized Hopf bifurcation.

```

for i=1:15
    plot(lpcbr.point(i).profile(1,:),...
         lpcbr.point(i).profile(2,:), 'Color', cm(1,:));
end
% Plot predicted periodic orbits
for i=1:7
    plot(lpcbr_pred.point(i).profile(1,:),...
         lpcbr_pred.point(i).profile(2,:), 'Color', cm(2,:));
end
xlabel('$u_1$', 'Interpreter', 'LaTeX');
ylabel('$u_2$', 'Interpreter', 'LaTeX');
title('Compare computed and predicted periodic orbits')
box on

```

The resulting plot is shown in [Figure S4](#). Note that the cycles shown have *different* underlying parameter values. Nonetheless, we see that the cycles are in good agreement.

S1.12 Simulation with MATLAB

Next, we simulate the dynamics near the generalized Hopf point. For this, we take a point in each of the three regions as shown in [Figure S3](#). The following code, from the file `FHN_simulation.m`, uses the MATLAB function `dde23`.

```

%% Clean workspace and add DDE-BifTool scripts
% to the MATLAB search path.
clear; % clear variables
close all; % close figures
ddebiftoolpath='../../../../';
addpath(strcat(ddebiftoolpath, 'ddebiftool'),...
        strcat(ddebiftoolpath, 'ddebiftool_extra_psol'),...
        strcat(ddebiftoolpath, 'ddebiftool_extra_nmf'),...
        strcat(ddebiftoolpath, 'ddebiftool_utilities'));

```

```

load('FHN_results.mat')
%% Point in region I
beta0=1.8779;
alpha0=-1.1001;
% Point near the steady-state
x1=0;
x2=0.01;
% Integrate
tfinal=1000;
sol = dde23(@(t,y,Z) funcs.sys_rhs([y,Z],...
    [beta0 alpha0 tau0]),tau0,[x1 x2],[0 tfinal]);
t=linspace(0,tfinal,1000);
y=deval(sol,t);
% Plot
title('Point in region I')
figure(1);clf;
xlabel('$u_1$', 'Interpreter', 'LaTeX')
ylabel('$u_2$', 'Interpreter', 'LaTeX')
plot(y(1,:),y(2,:))

%% Point in region II
beta0=1.9130;
alpha0=-0.9008;
% Point near the steady-state
x2=0.1;
% Integrate
sol = dde23(@(t,y,Z) funcs.sys_rhs([y,Z],[beta0 alpha0 tau0]),...
    tau0,[x1 x2],[0 tfinal]);
t=linspace(0,tfinal,1000);
y=deval(sol,t);
% Plot
figure(2);clf;
title('Point in region II')
xlabel('$u_1$', 'Interpreter', 'LaTeX')
ylabel('$u_2$', 'Interpreter', 'LaTeX')
plot(y(1,:),y(2,:))

%% Point in region III
beta0=1.8890;
alpha0=-0.6081;
% Orbit converging to periodic orbit
x2=0.1;
% Integrate
tfinal=3000; % use larger time interval
sol = dde23(@(t,y,Z) funcs.sys_rhs([y,Z],[beta0 alpha0 tau0]),...
    tau0,[x1 x2],[0 tfinal]);
t=linspace(0,tfinal,4000);
y1=deval(sol,t);
% Plot
figure(3);clf;
title('Point in region III')
xlabel('$u_1$', 'Interpreter', 'LaTeX')
ylabel('$u_2$', 'Interpreter', 'LaTeX')
plot(y1(1,:),y1(2,:))
% Orbit converging to the stable steady-state
% Integrate
x2=0.093;

```

```

sol = dde23(@(t,y,Z) funcs.sys_rhs([y,Z],[beta0 alpha0 tau0]),...
tau0,[x1 x2],[0 tfinal]);
y2=deval(sol,t);
% Add to plot
hold on; plot(y2(1,:),y2(2:,:), 'Color',cm(2,:));
%% Time series of the previous solutions in region III
figure(4);clf;
plot(t,y1(1,:),t,y2(1,:))
title('Time series of solutions in region III')
xlabel('$t$', 'Interpreter','LaTeX')
ylabel('$u_1$', 'Interpreter','LaTeX')

```

In [Figures S5a to S5d](#) the resulting plots are shown, confirming the dynamics near the generalized Hopf point as predicted in [\[43\]](#).

S2 Fold-Hopf bifurcation in the Rose–Hindmarsh model with time delay

In [\[48\]](#) a Rose-Hindmarsh model [\[33,34\]](#) with time delay in the self-feedback process,

$$\begin{cases} \dot{x}(t) = y(t) - ax^3(t) + bx^2(t - \tau) - cz(t) + I_{app}, \\ \dot{y}(t) = c - dx^2(t) - y(t), \\ \dot{z}(t) = r(S(x(t) - \chi) - z(t)), \end{cases} \quad (\text{S2})$$

is considered, see [Section 8.2](#). The parameters values

$$a = 1.0, \quad b = 3.0, \quad c = 1.0, \quad d = 5.0, \quad \chi = -1.6, \quad r = 0.001 \quad (\text{S3})$$

are fixed and (I_{app}, S) are the unfolding parameters.

Remark 24. This demonstration can be found in the directory `demos/tutorial/VII/RH` relative to the main directory of the DDE-BifTool package. Here, we omit the code to generate a system file. The system file `sym_RH_mf.m` has been generated with the script `gen_sym_RS.m`. Also, we assume that the DDE-BifTool package has been loaded as in [Listing S1](#). The code in [Sections S2.1 to S2.10](#) highlights the important parts of the file `RH.m`.

S2.1 Set parameter names and funcs structure

As in the previous example, we set the parameter names and define the `funcs` structure.

```

%% Set parameter names
parnames={'Iapp','S','r','tau'};
cind=[parnames;num2cell(1:length(parnames))];
ind=struct(cind{:});
%% Set funcs structure
% We load the precalculated multilinear forms. These have been
% generated with the file gen_sym_RH.m.
funcs=set_symfuncs(@sym_RH_mf,'sys_tau',@( )ind.tau);

```

S2.2 Stability and normal form coefficients of the fold-Hopf point

We construct a steady-state at the fold-Hopf point and calculate its stability.

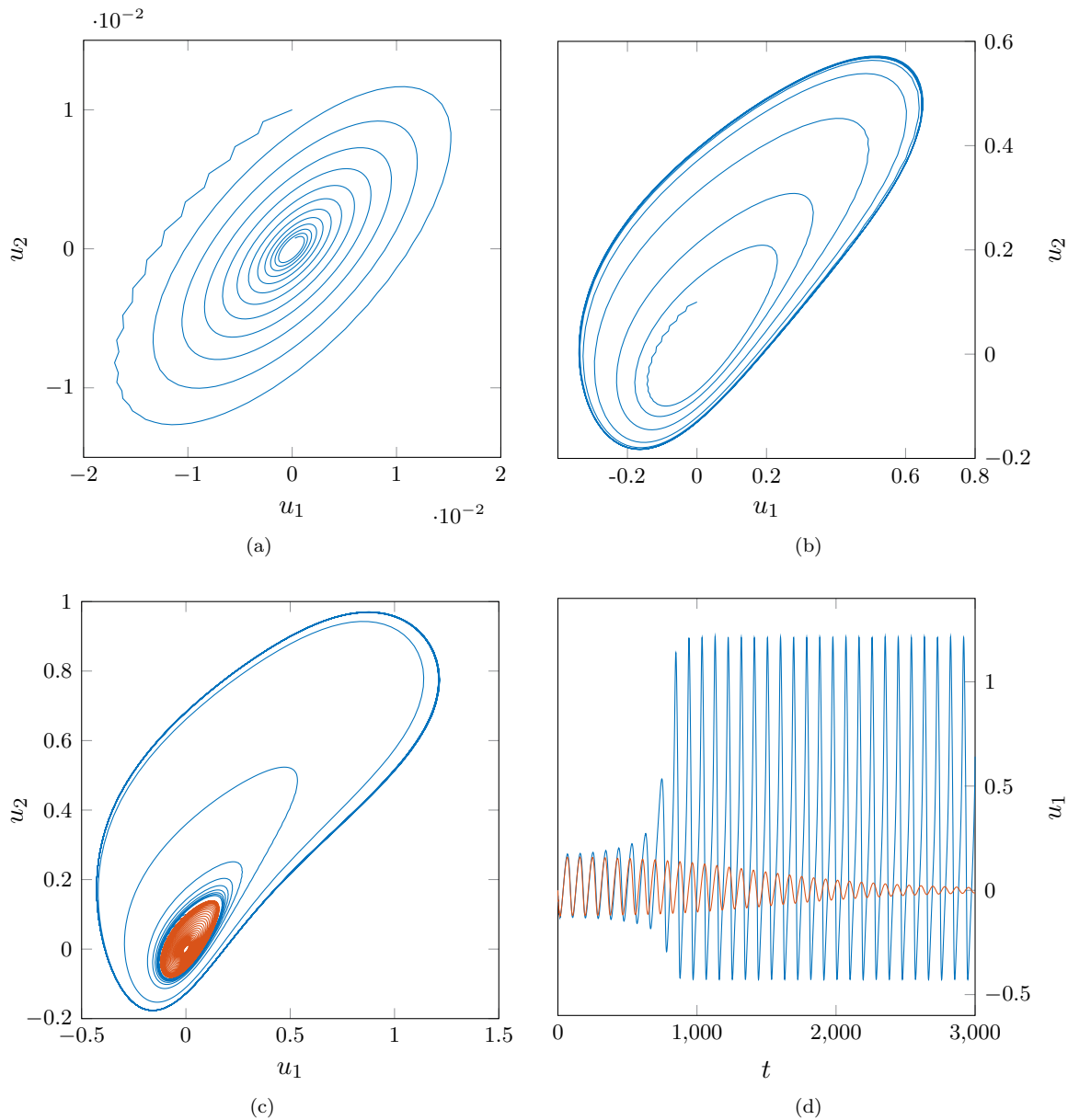


Figure S5: Simulation near the generalized Hopf point in the system (S1). In (a) we see a stable steady-state corresponding to a point in region I. When we enter region II, the stability of the steady-state is lost and a stable cycle appears, as seen in (b). In region III, there is a stable steady-state inside a stable cycle. This is confirmed in (c) and (d). In (c), the initial point of the orbit in blue is just outside the unstable cycle and converges to the stable cycles. The initial point of the orbit in red is just inside the unstable cycle and converges to the stable steady-state. In (d), the time series of these orbits are shown in the $(t, u_1(t))$ plane.

```

%% Construct fold-Hopf point
a=1.0; b=3.0; c=1.0; d=5.0; chi=-1.6;
r=1.0e-03;

```

```

S=-0.57452592;
[xstar,Iapp,tau]=bifurcationvalues(a,b,c,d,chi,r,S);
% Construct steady-state point
stst=dde_stst_create('x',[xstar; c-d*xstar^2; S*(xstar-chi)]);
stst.parameter([ind.Iapp ind.S ind.r ind.tau])=[Iapp S r tau];
% Calculate stability
method=df_mthod(funcs,'stst');
stst.stability=p_stabil(funcs,stst,method.stability);
stst.stability.l1(1:5)

```

The function `bifurcationvalues` calculates (x_*, I_{app}, τ) according to the formulas as given in [Section 8.2](#). The MATLAB console shows the following output.

```

ans =

    0.0000 + 0.0000i
   -0.0000 + 1.0079i
   -0.0000 - 1.0079i
   -0.0994 + 1.9324i
   -0.0994 - 1.9324i

```

We have a zero eigenvalue and a pair of purely imaginary eigenvalues. Furthermore, the remaining eigenvalues have negative real parts. Next, we calculate the normal form coefficients and the transformation to the center manifold with the function `nmfm_zeho`, which implements the coefficients as derived in [Section 6.2](#). For this we need to set the argument `free_pars` to the unfolding parameter (I_{app}, S) . These coefficients will be used to start the continuation of the various branches emanating from the fold-Hopf point.

```

% Calculate normal coefficients
hopf=p_tohopf(funcs,stst);
zeho=p_tozeho(hopf);
unfolding_pars=[ind.Iapp, ind.S];
zeho=nmfm_zeho(funcs,zeho,'free_pars',unfolding_pars);
zeho.nmfm

```

The MATLAB console shows the following output.

```

ans =

  struct with fields:

    g200: -0.0024
    g110: 0.3296 + 0.7006i
    g011: -0.0078
    g300: 0.0106
    g111: -0.0146
    g210: -2.7764 - 3.1806i
    g021: -0.3745 - 2.4754i
         b: -0.0024
         c: -0.0078
         d: 0.3296 + 5.2274i
         e: 15.6941
         s: 1.8487e-05
    theta: -139.0315
transcritical: 0
    h200: [1x1 struct]
    h011: [1x1 struct]

```

```

h020: [1x1 struct]
h110: [1x1 struct]
      K: [2x2 double]
h000mu: [1x2 struct]
omega1: 7.4540
omega2: 2.1259

```

Since $s > 0$ and $\theta(0) < 0$ global bifurcations or invariant tori are present. However, since the sign of e is positive the stability of the invariant tori will be unstable for nearby parameter values. It follows that the torus observed in [48] by simulations does not originate from the fold-Hopf point under consideration.

S2.3 Set bifurcation parameter range and step size bounds

Before continuing the various branches emanating from the fold-Hopf point, we create the variable `brpars` containing parameter bounds and maximal stepsizes.

```

% Set bifurcation parameter range and step size bounds
brpars={ 'min_bound',[ind.Iapp -20; ind.S -12],...
         'max_bound',[ind.Iapp 10; ind.S 5],...
         'max_step',[ind.Iapp 4e-02; ind.S 4e-02]};

```

S2.4 Continue NS, Hopf and fold branch

As in the previous example, we use the function `C1branch_from_C2point` to start to continue the branches emanating from the fold-Hopf point. [Figure S6](#) is created using similar code as in [Listing S3](#). We remark that even when there would be stable tori present for nearby parameter values, the window in which these tori would exist is quite small. Indeed, the parameter values would have to be below the Hopf curve and above the Neimark-Sacker curve to the left of the fold-Hopf point.

```

% Continue Neimark-Sacker curve emanating from fold-Hopf point
[trfuncs,nsbr,~]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
    'codim2',zeho.kind,'codim1','TorusBifurcation',...
    brpars{:},'step',1e-03,'plot',0);
ntrsteps=1000; [nsbr,suc]=br_contn(trfuncs,nsbr,ntrsteps);
% Continue Hopf curve emanating from fold-Hopf point
[~,hbr,suc]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
    'codim2',zeho.kind,'codim1','hopf',...
    brpars{:},'step',1e-03,'plot',0);
nop=1000; [hbr,suc]=br_contn(funcs,hbr,nop); assert(suc>0)
hbr=br_rvers(hbr);
[hbr,suc]=br_contn(funcs,hbr,nop);
% Continue fold curve emanating from fold-Hopf point
[~,fbr,suc]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
    'codim2',zeho.kind,'codim1','fold','step',1e-03,'plot',0);
nop=1000; [fbr,suc]=br_contn(funcs,fbr,nop);
fbr=br_rvers(fbr);
[fbr,suc]=br_contn(funcs,fbr,nop);

```

S2.5 Stable invariant tori

We change the parameters $r = 1.4$ and $S = -8$, while keeping the other fixed parameters as in [\(S3\)](#). Using the formulas given in [Section 8.2](#), we calculate x_* , I_{app} and τ .

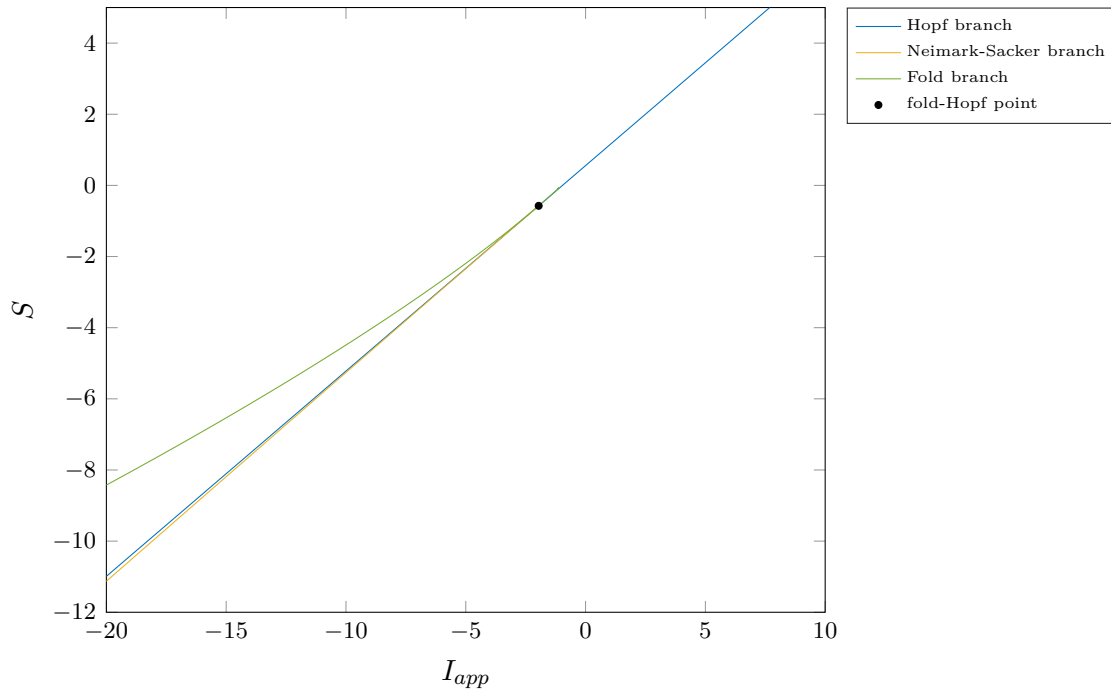


Figure S6: Bifurcation diagram near the fold-Hopf point in (S2) with $(r, S) = (0.001, -0.57452592)$.

```

%% Different parameters with stable torus
r=1.4; S=-8;
[xstar, Iapp, tau]=bifurcationvalues(a,b,c,d,chi,r,S);
% Construct steady-state point
stst=dde_stst_create('x',[xstar; c-d*xstar^2; S*(xstar-chi)]);
stst.parameter([ind.Iapp ind.S ind.r ind.tau])=[Iapp S r tau];
% Calculate stability
method=df_mthod(funcs,'stst');
stst.stability=p_stabil(funcs,stst,method.stability);
stst.stability.l1(1:5)

```

The MATLAB console outputs

```

ans =

    0.00000 + 0.00000i
    0.00000 + 5.60424i
    0.00000 - 5.60424i
   -0.27870 + 0.00000i
   -0.66607 + 11.94839i
   -0.66607 - 11.94839i

```

which are indeed the eigenvalues that should be present at a fold-Hopf bifurcation. Next, we calculate the normal form coefficients.

```

% Calculate normal coefficients
hopf=p_tohopf(funcs,stst);
zeho=p_tozeho(hopf);
unfolding_pars=[ind.Iapp, ind.S];

```

```
zeho=nmfm_zeho(funcs,zeho,'free_pars',unfolding_pars);
```

Inspecting the normal form coefficients yields

```
>> fprintf('s=%f, theta=%f, e=%f\n',...
          zeho.nmfm.s,zeho.nmfm.theta,zeho.nmfm.e)
```

```
s=1.770013, theta=-0.156886, e=-0.037794
>>
```

The coefficients s and $\theta(0)$ reveal that we are in case III of the fold-Hopf bifurcation, see [43, page 342]. Since the sign of $e(0)$ is negative, there is a time reversal to take into account. Therefore, we expect stable tori to be present for nearby parameter values.

S2.6 Adjusting bifurcation parameter range

We adjust the variable `brpars` to reflect the current situation.

```
% Set bifurcation parameter range and step size bounds
brpars={'min_bound',[ind.Iapp -20; ind.S -9],...
        'max_bound',[ind.Iapp -18; ind.S -7],...
        'max_step',[ind.Iapp 0.04; ind.S 0.04]};
```

S2.7 Detect special points on the Hopf branch

Since the code to continue the Neimark-Sacker, Hopf and fold curves is identical to the code in [Section S2.4](#), we continue with detecting bifurcations on the Hopf branch. The Hopf points on the branch `hbr_wbifs` will contain the normal form coefficients `L1` and `L2`. These will be used to visualize the criticality of the Hopf points (sub or super) in the bifurcation diagram.

```
[hbr_wbifs,hopftests,hc2_indices,hc2_types]=...
  LocateSpecialPoints(funcs,hbr);
```

S2.8 Predictors

As in the previous example, we obtain predictors for the various branches simply by setting the argument `predictor` to 1 and the argument `step` to a range of ε -values when calling the function `C1branch_from_C2point`.

```
% Predictors for Neimark-Sacker and Hopf curves
[~,nsbr_pred]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
  'codim2','zeho','codim1','TorusBifurcation',...
  'step',linspace(1e-03,2.2e-01,40),'predictor',1);
[~,hbrsub_pred]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
  'codim2','zeho','codim1','hopf',...
  'step',linspace(0,1e-03,20),'predictor',1);
[~,hbrsup_pred]=C1branch_from_C2point(funcs,zeho,unfolding_pars,...
  'codim2','zeho','codim1','hopf',...
  'step',linspace(-1e-03,0,20),'predictor',1);
```

S2.9 Bifurcation diagram

We plot the obtained curves and the predictor for the Neimark-Sacker and Hopf curve with the following code.

```

%% Plot comparing computed and predicted Neimark-Sacker curve
figure(8); clf; hold on;
nsbr2_pm_pred = [getpars(nsbr_pred,ind.Iapp); ...
    getpars(nsbr_pred,ind.S)];
hbrsub_pm_pred = [getpars(hbrsub_pred,ind.Iapp); ...
    getpars(hbrsub_pred,ind.S)];
hbrsup_pm_pred = [getpars(hbrsuper_pred,ind.Iapp); ...
    getpars(hbrsuper_pred,ind.S)];
plot(hbrsub_pm(1,:),hbrsub_pm(2:),'Color',cm(1,:),...
    'DisplayName','subcritical Hopf branch');
plot(hbrsuper_pm(1,:),hbrsuper_pm(2:),'Color',cm(2,:),...
    'DisplayName','supercritical Hopf branch');
plot(hbrsub_pm_pred(1,:),hbrsub_pm_pred(2:),'.','Color',cm(1,:),...
    'DisplayName','subcritical Hopf predictor');
plot(hbrsup_pm_pred(1,:),hbrsup_pm_pred(2:),'.','Color',cm(2,:),...
    'DisplayName','supercritical Hopf predictor');
plot(nsbr1_pm(1,:),nsbr1_pm(2:),'Color',cm(3,:),...
    'DisplayName','Neimark-Sacker branch');
plot(nsbr2_pm_pred(1,:),nsbr2_pm_pred(2:),'.','Color',cm(3,:),...
    'DisplayName','Neimark-Sacker predictor');
plot(zeho.parameter(ind.Iapp),zeho.parameter(ind.S),'k.',...
    'MarkerSize',12,'DisplayName','fold-Hopf point')
title('Neimark-Sacker curve emanating from the fold-Hopf point')
axis([-19.0193 -18.7128 -8.0477 -7.9587])
xlabel('$I_{app}$','Interpreter','LaTeX');
ylabel('$S$','Interpreter','LaTeX');
text(-18.784,-7.981, 'I','FontSize',14); % stable period orbit
text(-18.905,-8.032, 'II','FontSize',14); % stable 2d torus
legend('Location','NorthWest'); box on

```

Figure S7 shows the resulting bifurcation diagram.

S2.10 Plots comparing computed and predicted periodic orbits

We create a plot to compare the computed and predicted periodic orbits.

```

%% Plot comparing computed and predicted periodic orbits
figure(9); clf; hold on;
for i=1:14
    plot3(nsbr.point(i).profile(1,:),nsbr.point(i).profile(2,:),...
        nsbr.point(i).profile(3:),'Color',cm(1,:));
end
for i=1:9
    plot3(nsbr_pred.point(i).profile(1,:),...
        nsbr_pred.point(i).profile(2,:),...
        nsbr_pred.point(i).profile(3:),'Color',cm(2,:));
end
title('Comparison between computed and predicted periodic orbits')
xlabel('x'); ylabel('y'); zlabel('z')
view(3); grid on

```

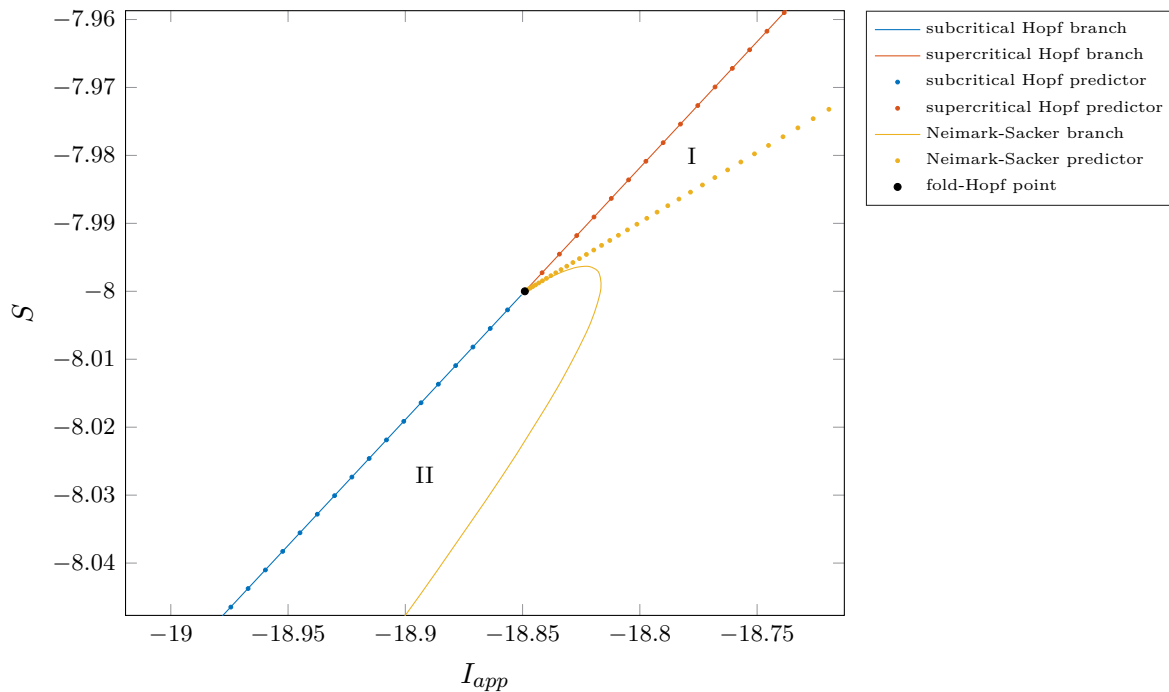


Figure S7: Bifurcation diagram near the fold-Hopf point in (S2) with $(r, S) = (1.4, -8)$. The fold branch is not included here since it is indistinguishable from the Hopf curve at this scale.

The resulting plot is shown in Figure S8a. To compare the computed and predicted periods, we use the following code.

```

%% Compare computed and predicted periods
figure(10); clf; hold on;
% Plot computed periods on nsbr(1) and nsbr(2)
omegas1=arrayfun(@(p)p.period,nsbr.point);
omegas1_pred=arrayfun(@(p)p.period,nsbr_pred.point);
plot(getpars(nsbr,ind.Iapp),omegas1,'Color',cm(1,:));
plot(getpars(nsbr_pred,ind.Iapp),omegas1_pred,'.','Color',cm(1,:));
title('Compare computed and predicted periods')
xlabel('$I_{app}$','Interpreter','LaTeX')
ylabel('$\omega$','Interpreter','LaTeX')
axis([-19.1016 -18.6500 1.1144 1.1220])
legend('Computed period','Predicted period','Location','SouthEast')

```

The resulting plot is shown in Figure S8b.

S2.11 Simulation with pydelay

In this section we simulate the dynamics near the fold-Hopf point. The following code, from the file `RH_simulation_torus.py`, uses the Python package `pydelay` [24].

```

import numpy as np
from pydelay import dde23
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

```

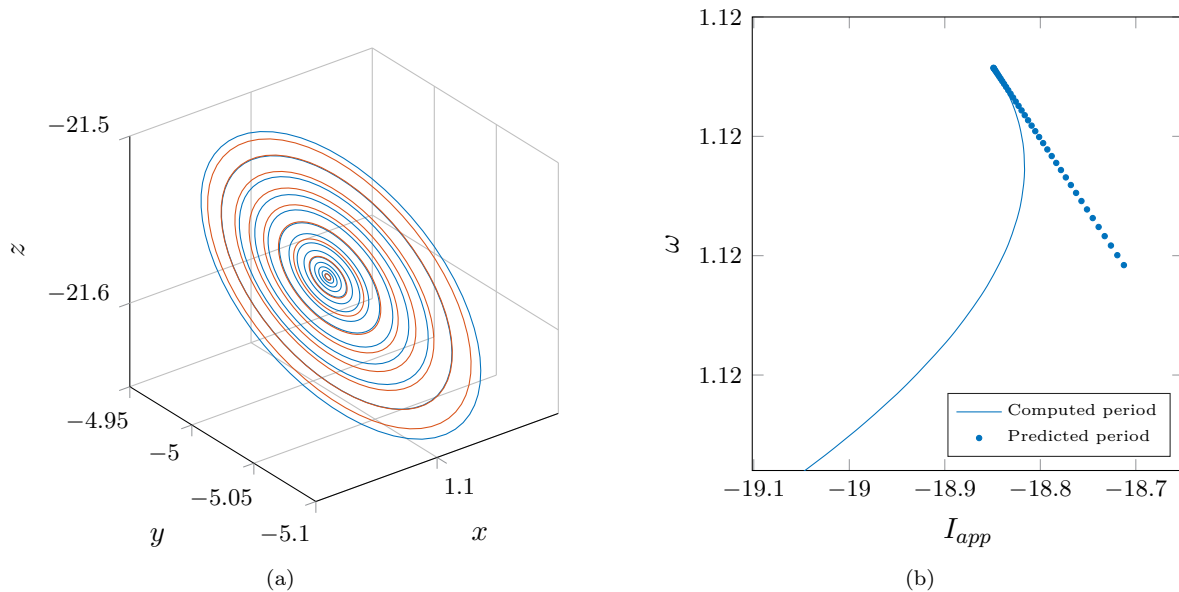


Figure S8: In (a) the computed periodic orbits (blue) and predicted periodic orbits (red) are compared. In (b) the computed predicted periods of the cycles are compared. We see that both are in good agreement.

```
import matplotlib.pyplot as plt
from matplotlib import colors

# Number of time units
tfinal_cycle = 10000
tfinal_torus = 20000

# Define DDE
eqns = {
    'x' : 'y-a*pow(x,3)+b*pow(x(t-tau),2)-c*z+Iapp',
    'y' : 'c-d*pow(x,2)-y',
    'z' : 'r*(S*(x-chi)-z)'
}

# Set parameters for torus
params_torus = {
    'a':1, 'b':3, 'c':1, 'd':5.0, 'chi':-1.6, 'r':1.4,
    'tau':0.940246941050084,
    'Iapp':-18.902420391705071,
    'S':-8.045234985422740
}

# Set parameters for torus
params_cycle = params_torus.copy()
params_cycle['Iapp']=-18.886177304147466
params_cycle['S']=-8.044197084548104

# Set number of timesteps from the end to plot
```



```

timesteps_torus=300
timesteps_cycle=10

# Select periodic orbit or torus
tfinal, params, timesteps=tfinal_cycle, params_cycle, timesteps_cycle
#tfinal, params, timesteps=tfinal_torus, params_torus, timesteps_torus

# Solve DDE
dde = dde23(eqns=eqns, params=params)
dde.set_sim_params(tfinal=tfinal)
dde.set_sim_params(tfinal=tfinal, dtmax=0.001)
histfunc = {
    'x': lambda t: 1.097167540709727,
    'y': lambda t: -5.018883061935152,
    'z': lambda t: -21.577340325677817
}
dde.hist_from_funcs(histfunc, 51)
dde.run()

# Subtract solution components
sol0 = dde.sample(tfinal-timesteps, tfinal, 0.001)
t = sol0['t']
x = sol0['x']
y = sol0['y']
z = sol0['z']

# Plot time series
fig = plt.figure()
plt.figure(1)
plt.subplot(311)
plt.xlabel('$t$')
plt.ylabel('$x(t)$')
plt.plot(t, x, c='royalblue')

plt.subplot(312)
plt.xlabel('$t$')
plt.ylabel('$y(t)$')
plt.plot(t, y, c='royalblue')

plt.subplot(313)
plt.xlabel('$t$')
plt.ylabel('$z(t)$')
plt.plot(t, z, c='royalblue')
fig.set_size_inches(18.5, 10.5)
plt.show()

# Plot the solution in phase-space
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.w_xaxis.set_panel_color((1.0, 1.0, 1.0, 1.0))
ax.w_yaxis.set_panel_color((1.0, 1.0, 1.0, 1.0))
ax.w_zaxis.set_panel_color((1.0, 1.0, 1.0, 1.0))
ax.plot(x, y, z, c='royalblue')

ax.set_xlabel('$x(t)$')

```

```

ax.set_ylabel('$y(t)$')
ax.set_zlabel('$z(t)$')

# Fix to get z_label inside the figure
from matplotlib import rcParams
rcParams.update({'figure.autolayout': True})

ax.view_init(9, -44)
plt.show()

# Subtract solution components for Poincaré section
sol0 = dde.sample(tfinal-1000, tfinal, 0.001)
t = sol0['t']
x = sol0['x']
y = sol0['y']
z = sol0['z']

# Poincaré section
def poincaresection(x1, x2, x3, x1_label, x2_label, val):
    zero_cross = np.where(np.diff(np.sign(x3-val)))
    plt.figure(1)
    plt.xlabel(x1_label)
    plt.ylabel(x2_label)
    plt.plot(x1[zero_cross], x2[zero_cross], '.', c='royalblue')
    plt.show()
    return

x1_label='$x(t)$'
x2_label='$y(t)$'
poincaresection(x,y,z,x1_label,x2_label,-21.75)

```

We will simulate the dynamics in region I and II, see [Figure S7](#), where a stable periodic orbit and stable two-dimensional torus, respectively, should be present. For a point in region I, we will take the unfolding parameter values

$$(I_{app}, S) = (-18.886177304147466, -8.044197084548104).$$

As an initial condition, we use the constant history function with values of the location of the fold-Hopf point. We integrate the DDE on the time interval $t \in [0, 10.000]$ with `pydelay`. In [Figure S9](#) the time series of the components x, y and z and the orbit in (x, y, z) -space of the last 10 time steps are shown, clearly indicating a stable orbit.

Next, we simulate the dynamics in region II. Therefore, we adjust the unfolding parameter values to

$$(I_{app}, S) = (-18.902420391705071, -8.045234985422740).$$

Furthermore, we increase the integration interval to $t \in [0, 20.000]$. Keeping the history function the same, we plot the last 1000 time steps, see [Figure S9](#). We conclude that the dynamics near the fold-Hopf point are as predicted in [\[43\]](#).

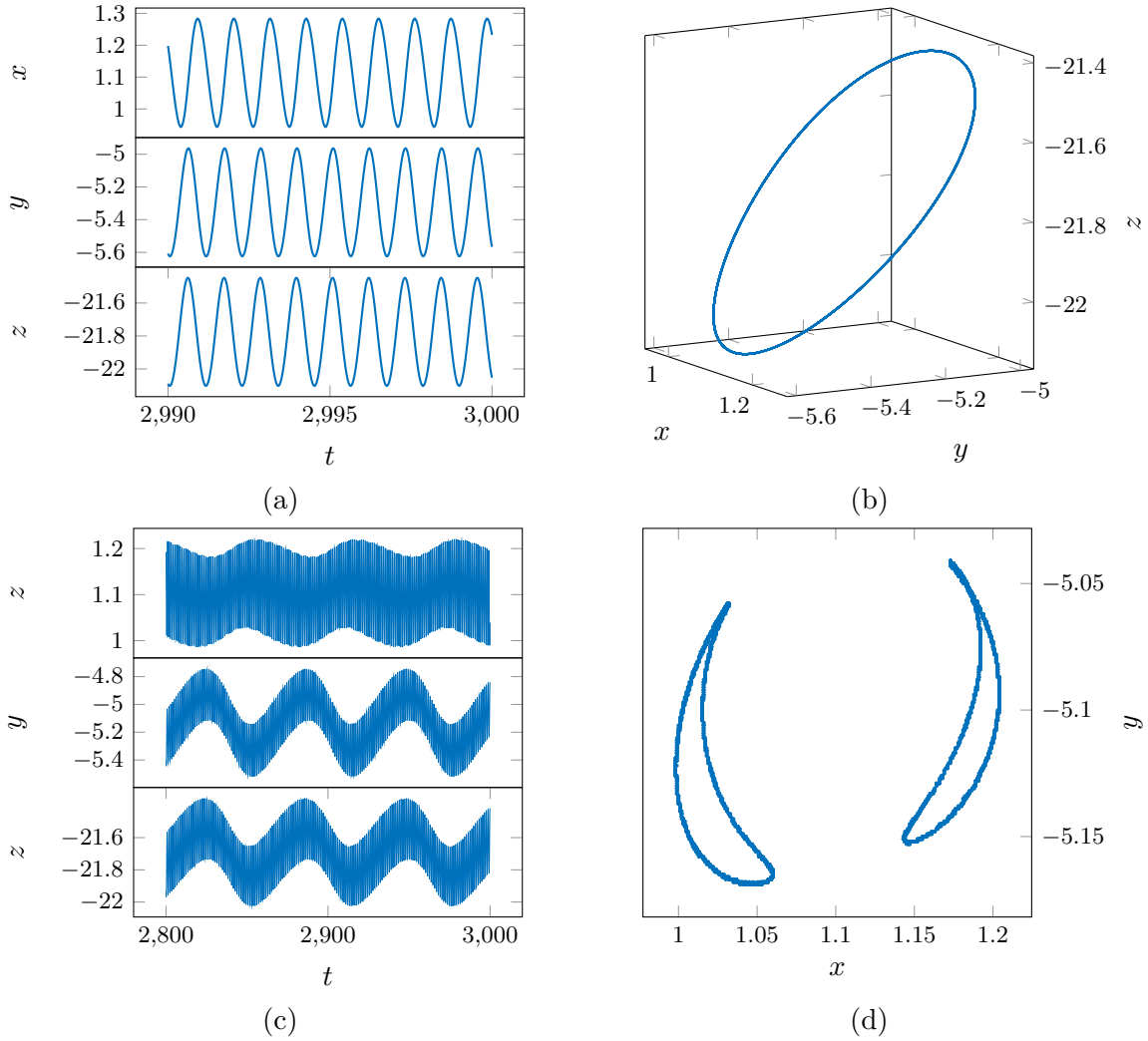


Figure S9: Simulation near fold-Hopf point in (102). In (a) and (b), a stable periodic solution is shown. In (c) and (d), a stable torus is shown. In (c), the time series is plotted while in (d), the cross-section defined by $z(t) = -21.75$ in the phase-space (x, y, z) is taken.

S3 Hopf-Hopf and generalized Hopf bifurcations in Active control system

In Section 8.3 we considered the following active control system

$$\begin{cases} \dot{x}(t) = \tau y(t), \\ \dot{y}(t) = \tau(-x(t) - g_u x(t-1) - 2\zeta y(t) - g_v y(t-1) + f(t)), \end{cases} \quad (\text{S4})$$

which is used to control the response of structures to internal or external excitation, see [50]. The function f is substituted by $\beta x^3(t-1)$ and the parameters

$$g_u = 0.1, \quad g_v = 0.52, \quad \beta = 0.1,$$

are fixed. The control parameters are ζ and τ .

Remark 25. This demonstration can be found in the directory `demos/tutorial/VII/acs` relative to the main directory of the DDE-BifTool package. Here, we omit the code to generate a system file. The system file `sym_acs_mf.m` has been generated with the script `gen_acs.m`. Also, we assume that the DDE-BifTool package has been loaded as in [Listing S1](#). The code in [Sections S3.1](#) to [S3.10](#) highlights the important parts of the file `acs.m`.

S3.1 Set parameter names and funcs structure

We set the parameter names and define the `funcs` structure.

```
%% Set parameter names
parnames={'zeta','tau','tau_scaled'};
cind=[parnames;num2cell(1:length(parnames))];
ind=struct(cind{:});
%% Set funcs structure
% We load the precalculated multilinear forms. These have been
% generated with the file gen_sym_acs.m.
funcs=set_symfuncs(@sym_acs_mf,'sys_tau',@()ind.tau_scaled);
```

S3.2 Stability and normal form coefficients of the Hopf-Hopf point

We construct a steady-state at the Hopf-Hopf point and calculate its stability.

```
%% Hopf-Hopf point
% Construct steady-state point
stst.kind='stst';
stst.x=[0;0];
stst.parameter(ind.zeta)=-0.016225;
stst.parameter(ind.tau)=5.89802;
stst.parameter(ind.tau_scaled)=1;
% Calculate stability
method=df_mthod(funcs,'stst');
stst.stability=p_stabil(funcs,stst,method.stability);
stst.stability.11
```

The MATLAB console shows the following output.

```
ans =

    0.0000 + 4.5275i
    0.0000 - 4.5275i
   -0.0000 + 7.6449i
   -0.0000 - 7.6449i
```

The eigenvalues confirm that the point under consideration is indeed a Hopf-Hopf point. Furthermore, the remaining eigenvalues have negative real parts. Next, we calculate the normal form coefficients and the transformation to the center manifold with the function `nmfm_hoho`, which implements the coefficients as derived in [Section 6.3](#). For this we need to set the argument `free_pars` to the unfolding parameter (ζ, τ) . These coefficients will be used to start the continuation of the various branches emanating from the Hopf-Hopf point.

```
%% Calculate coefficients of the parameter dependent normal form
hopf=p_tohopf(funcs,stst);
method=df_mthod(funcs,'hopf');
```

```
hopf.stability=p_stabil(funcs,hopf,method.stability);
hoho=p_tohoho(hopf);
unfolding_parameters=[ind.zeta, ind.tau];
hoho=nmfm_hoho(funcs,hoho,'free_pars',unfolding_parameters);
hoho.nmfm
```

The MATLAB console shows the following output.

```
ans =
```

```
struct with fields:
```

```
g2100: -0.0915 + 0.1214i
g1011: -0.3084 + 0.4096i
g1110: 0.2151 + 0.3876i
g0021: 0.1813 + 0.3268i
theta: -1.7009
delta: -2.3517
      b: [2x2 double]
h0011: [1x1 struct]
h0020: [1x1 struct]
h2000: [1x1 struct]
      K: [2x2 double]
h0000: [1x2 struct]
```

We conclude that this Hopf-Hopf bifurcation is of ‘difficult’ type, since

$$(\operatorname{Re} g_{2100})(\operatorname{Re} g_{0021}) = -0.0166 < 0,$$

see [43]. Furthermore, the quantities

$$\theta = \frac{\operatorname{Re} g_{1011}}{\operatorname{Re} g_{0021}} = -1.7009, \quad \delta = \frac{\operatorname{Re} g_{1101}}{\operatorname{Re} g_{2100}} = -2.3517$$

are such that $\theta < 0$, $\delta < 0$, $\theta\delta > 0$. It follows that we are in case VI.

S3.3 Set bifurcation parameter range and step size bounds

Before continuing the various branches emanating from the transcritical-Hopf point, we create the variable `brpars` containing parameter bounds and maximal stepsizes.

```
% Set bifurcation parameter range and step size bounds
brpars={'max_bound',[ind.tau 16],...
       'min_bound',[ind.tau 5],...
       'max_step',[ind.zeta 0.04; ind.tau 0.04]};
```

S3.4 Continuing Hopf and Neimark-Sacker bifurcation curves

We use the function `C1branch_from_C2point` to start to continue the branches emanating from the Hopf-Hopf point.

```
% Continue Neimark-Sacker curves emanating from Hopf-Hopf point
[trfuncs,nsbr,suc]=C1branch_from_C2point(funcs,hoho,...
    unfolding_parameters,'codim2','hoho','codim1',...
    'TorusBifurcation',brpars{:},'step',1e-01,'plot',0);
assert(all(suc(:)>0))
```

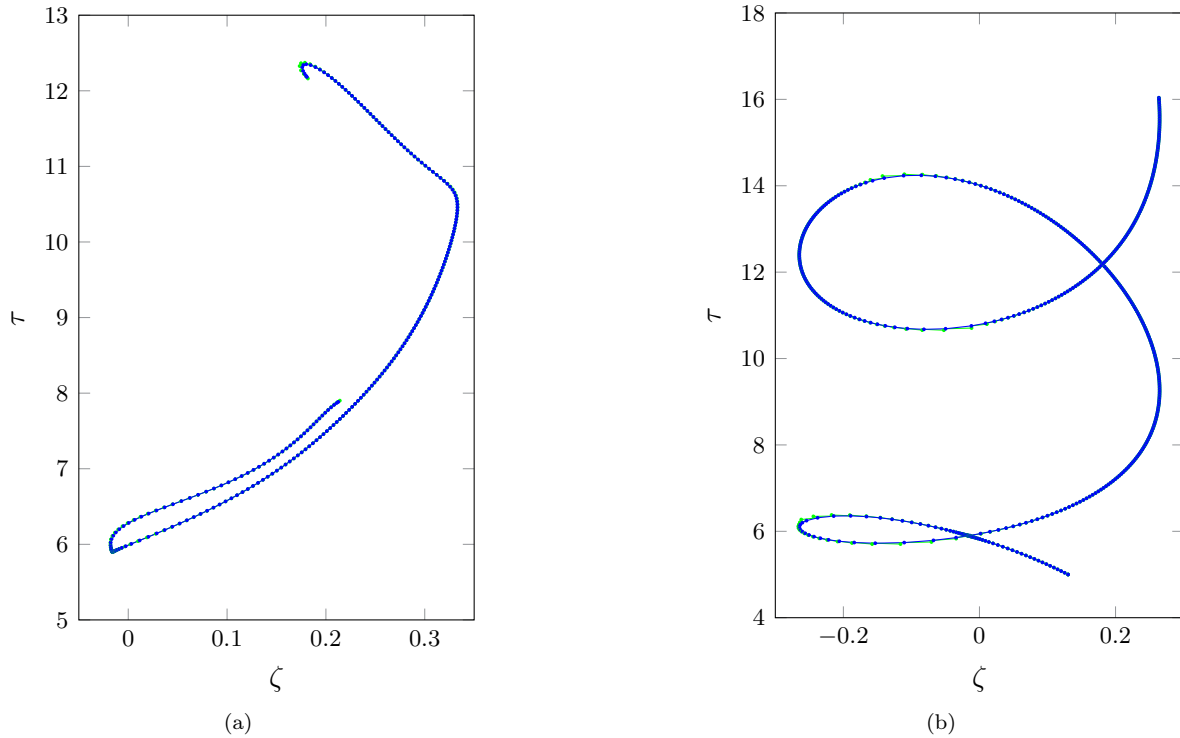


Figure S10: In (a), the continued branches `ns1_br` and `ns2_br` are plotted. In (b), the continued Hopf branch `hopf_br1` is plotted.

```
ntrsteps=186; [nsbr(1),suc]=br_contn(trfuncs,nsbr(1),ntrsteps);
assert(suc>0)
ntrsteps=61; [nsbr(2),suc]=br_contn(trfuncs,nsbr(2),ntrsteps);
assert(suc>0)
```

```
% Continue Hopf curve emanating from Hopf-Hopf point
[~,hbr,suc]=C1branch_from_C2point(funcs,hoho,...
    unfolding_parameters,'codim2','hoho','codim1','hopf'....
    ,brpars{:},'step',1e-03,'plot',0);
assert(all(suc(:)>0))
nop=1000; [hbr(1),suc]=br_contn(funcs,hbr(1),nop);assert(suc>0)
hbr(1)=br_rvers(hbr(1));
[hbr(1),suc]=br_contn(funcs,hbr(1),nop); assert(suc>0)
nop=10; [hbr(2),suc]=br_contn(funcs,hbr(2),nop); assert(suc>0)
hbr(2)=br_rvers(hbr(2));
[hbr(2),suc]=br_contn(funcs,hbr(2),nop); assert(suc>0)
```

In **Figure S10**, the computed branches are shown. We see in **Figure S10b** that it is redundant to continue the second Hopf branch emanating from the Hopf-Hopf point. Indeed, the first Hopf branch connects the Hopf-Hopf point to itself. Here, we verified that the underlying points coincide.

S3.5 Predictors

For comparison in the bifurcation diagram, we obtain predictors for the various branches by setting the argument `predictor` to 1 and `step` to a range of ε -values when calling the function `C1branch_from_C2point`.

```
%% Predictors for Neimark-Sacker and Hopf curves
[trfuncs,nsbr_pred,suc]=C1branch_from_C2point(funcs,hoho,...
    unfolding_parameters,'codim2','hoho','codim1',...
    'TorusBifurcation','step',linspace(1e-03,2,40),'predictor',1);
[~,hbr_pred,suc]=C1branch_from_C2point(funcs,hoho,...
    unfolding_parameters,'codim2','hoho','codim1','hopf',...
    'step',linspace(-2e-01,2e-01,30),'predictor',1);
```

S3.6 Bifurcation diagram

We plot the computed curves and the predictors for the Neimark-Sacker and Hopf curves with the following code.

```
%% Close-up near Hopf Hopf point in parameter space with predictors
figure(2); clf; hold on;
hbr1_pred_pm = [getpars(hbr_pred(1), ind.zeta)
    getpars(hbr_pred(1), ind.tau)];
hbr2_pred_pm = [getpars(hbr_pred(2), ind.zeta)
    getpars(hbr_pred(2), ind.tau)];
nsbr1_pred_pm = [getpars(nsbr_pred(1),ind.zeta)
    getpars(nsbr_pred(1),ind.tau)];
nsbr2_pred_pm = [getpars(nsbr_pred(2),ind.zeta)
    getpars(nsbr_pred(2),ind.tau)];
plot(hbr_pm(1,:),hbr_pm(2:),'Color',cm(1,:),...
    'DisplayName','Hopf branches')
plot(nsbr1_pm(1,:),nsbr1_pm(2:),'Color',cm(2,:),...
    'DisplayName','Neimark-Sacker branches')
plot(nsbr2_pm(1,:),nsbr2_pm(2:),'Color',cm(2,:),...
    'HandleVisibility','off')
plot(hbr1_pred_pm(1,:), hbr1_pred_pm(2,:) ,'.',...
    'Color',cm(1:),'DisplayName','Hopf predictors')
plot(hbr2_pred_pm(1,:), hbr2_pred_pm(2,:) ,'.',...
    'Color',cm(1:),'HandleVisibility','off')
plot(nsbr1_pred_pm(1,:),nsbr1_pred_pm(2:),'.',...
    'Color',cm(2:),'DisplayName','Neimark-Sacker predictors')
plot(nsbr2_pred_pm(1,:),nsbr2_pred_pm(2:),'.',...
    'Color',cm(2:),'HandleVisibility','off')
plot(hoho.parameter(ind.zeta),hoho.parameter(ind.tau),'k.',...
    'MarkerSize',12,'DisplayName','Hopf Hopf point')
title(['Close-up near Hopf Hopf point '...
    'in parameter space with predictors'])
xlabel('$\zeta$', 'Interpreter', 'LaTeX');
ylabel('$\tau$', 'Interpreter', 'LaTeX');
legend('Location', 'NorthWest')
axis([-0.0562 0.0109 5.7098 6.1757])
box on
```

In [Figure S11a](#) the predictors in parameter space are compared.

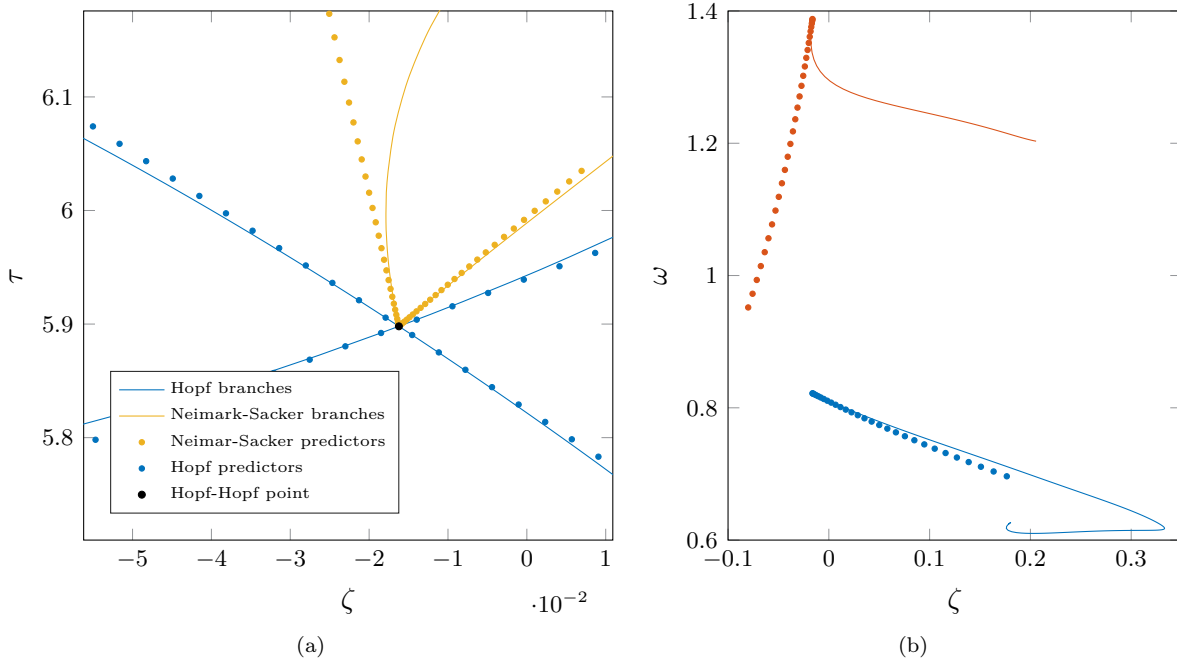


Figure S11: In (a), the computed curves are compared with the predicted curves for the Hopf-Hopf point `hoho`. In (b), the computed and predicted periods of the cycles are compared.

S3.7 Plot comparing computed and predicted periods

To compare the computed and predicted periods, we use the following code.

```

%% Compare computed and predicted periods
figure(3); clf; hold on;
% Plot computed periods on nsbr(1) and nsbr(2)
omegas1=arrayfun(@(p)p.period,nsbr(1).point);
omegas2=arrayfun(@(p)p.period,nsbr(2).point);
omegas1_pred=arrayfun(@(p)p.period,nsbr_pred(1).point);
omegas2_pred=arrayfun(@(p)p.period,nsbr_pred(2).point);
plot(getpars(nsbr(1),ind.zeta),omegas1,'Color',cm(1,:));
plot(getpars(nsbr(2),ind.zeta),omegas2,'Color',cm(2,:));
plot(getpars(nsbr_pred(1),ind.zeta),...
      omegas1_pred, '.', 'Color', cm(1,:));
plot(getpars(nsbr_pred(2),ind.zeta),...
      omegas2_pred, '.', 'Color', cm(2,:));
title('Compare computed and predicted periods')
xlabel('$\zeta$', 'Interpreter', 'LaTeX');
ylabel('$\omega$', 'Interpreter', 'LaTeX');

```

In [Figure S11b](#), the computed and predicted periods of the cycles are compared using the formulas as given in [Section 5.3.2](#).

S3.8 Detect special points on the Hopf branch

Using the detection capabilities from DDE-BifTool via the function `LocateSpecialPoints`, we detect one additional Hopf-Hopf point and three generalized Hopf points.


```

%% Detect codimension two points on hopf_br1
[hopf_br_wbifs, hopftests, hc2_indices, hc2_types]=...
  LocateSpecialPoints(funcs, hbr(1));

```

The MATLAB console shows the following output.

```

HopfCodimension2: calculate stability if not yet present
HopfCodimension2: calculate L1 coefficients
HopfCodimension2: (provisional) 3 gen. Hopf 4 Hopf-Hopf detected.
br_insert: detected 1 of 7: hoho. Normalform:
  g2100: 0.1813 + 0.3268i
  g1011: 0.2151 + 0.3876i
  g1110: -0.3084 + 0.4096i
  g0021: -0.0915 + 0.1214i
  theta: -2.3517
  delta: -1.7009

br_insert: detected 2 of 7: genh. Normalform:
  L2: -0.0458
  L1: -1.6706e-11

br_insert: detected 3 of 7: hoho. Normalform:
  g2100: 0.1813 + 0.3268i
  g1011: 0.2151 + 0.3876i
  g1110: -0.3084 + 0.4096i
  g0021: -0.0915 + 0.1214i
  theta: -2.3517
  delta: -1.7009

br_insert: detected 4 of 7: hoho. Normalform:
  g2100: -0.0034 + 0.3217i
  g1011: -0.0046 + 0.4380i
  g1110: -0.1494 + 0.4498i
  g0021: -0.0509 + 0.1531i
  theta: 0.0912
  delta: 43.8639

br_insert: detected 5 of 7: genh. Normalform:
  L2: 0.0070
  L1: -5.9006e-14

br_insert: detected 6 of 7: genh. Normalform:
  L2: 0.0075
  L1: 2.2535e-14

br_insert: detected 7 of 7: hoho. Normalform:
  g2100: -0.0034 + 0.3217i
  g1011: -0.0046 + 0.4380i
  g1110: -0.1494 + 0.4498i
  g0021: -0.0509 + 0.1531i
  theta: 0.0912
  delta: 43.8639

```

There are seven bifurcations detected on the Hopf branch `hbr(1)`: four Hopf-Hopf bifurcations and three generalized Hopf bifurcations. However, by inspecting the parameters of the detected Hopf Hopf points suggest that there are only two distinct Hopf-Hopf points, which are connected with the same Hopf

branch. By comparing the location of the underlying points confirms the premise. We are therefore not interested in continuing the Hopf curves emanating from these points. What we are interested in is continuing the Neimark-Sacker and limit point of cycle curves emanating from the second Hopf-Hopf point and the three generalized Hopf points, respectively.

The normal form coefficients of the second Hopf-Hopf point are such that

$$(\operatorname{Re} g_{2100})(\operatorname{Re} g_{0021}) = 1.7331e - 04 > 0$$

and

$$\theta \geq \delta > 0, \quad \delta\theta = 4 > 1.$$

We conclude that we are in case I of the ‘simple’ type, see [46, page 360]. Therefore, no stable invariant two-dimensional torus is predicted for nearby parameter values. We only expect to find two stable periodic orbits.

S3.9 Continuing third Neimark-Sacker bifurcation and limit point of cycle bifurcation curves

It turns out that we only need to continue one of the Neimark-Sacker bifurcation curves emanating from the second Hopf-Hopf point. Indeed, the other Neimark-Sacker bifurcation curve is given by `ns_br(1)`.

```

%% Subtract generalized Hopf points
genh_indices=hc2_indices(strcmp(hc2_types, 'genh'));
genhpts=hopf_br_wbifs.point(genh_indices);

%% Continue limit point of cycles emanating from the generalized Hopf points
[~,lpc_br1,suc]=C1branch_from_C2point(funcs,genhpts(1),...
unfolding_parameters,'codim2','genh','codim1','P0fold',...
brpars{:},'step',1e-01,'plot',0);
assert(all(suc>0))
[poffuncs,lpc_br2,suc]=C1branch_from_C2point(funcs,genhpts(3),...
unfolding_parameters,'codim2','genh','codim1','P0fold',...
brpars{:},'step',5e-03,'plot',0);
assert(all(suc>0))
ntrsteps=193; [lpc_br1,suc]=br_contn(poffuncs,lpc_br1,ntrsteps);
assert(suc>0)
ntrsteps=220; [lpc_br2,suc]=br_contn(poffuncs,lpc_br2,ntrsteps);
assert(suc>0)

%% Subtract second Hopf-Hopf point on the Hopf branch
hoho_indices=hc2_indices(strcmp(hc2_types, 'hoho'));
hoho2=hopf_br_wbifs.point(hoho_indices(3));

%% Continue Neimark-Sacker curves emanating from Hopf-Hopf point
[trfuncs,nsbr34,suc]=C1branch_from_C2point(funcs,hoho2,...
unfolding_parameters,'codim2','hoho','codim1',...
'TorusBifurcation',brpars{:},'step',5e-03,'plot',0);
assert(all(suc(:)>0))
ntrsteps=200; [nsbr34(2),suc,fail,rjct]=...
br_contn(trfuncs,nsbr34(2),ntrsteps);

```

S3.10 Bifurcation diagram

We plot the computed degenerate Hopf points, the limit point of limit cycle curves, and the Neimark-Sacker curves with the following code.

```

%% Bifurcation diagram in  $(\zeta, \tau)$ 
figure(4); clf; hold on;
% Subtract parameter values from the branches
lpc_br1_pm= [getpars(lpc_br1, ind.zeta)
             getpars(lpc_br1, ind.tau)];
lpc_br2_pm= [getpars(lpc_br2, ind.zeta)
             getpars(lpc_br2, ind.tau)];
nsbr3_pm   = [getpars(nsbr34(2), ind.zeta)
             getpars(nsbr34(2), ind.tau)];
% Plot curves
plot(hbr_pm(1,:), hbr_pm(2,:), 'Color',cm(1,:),...
     'DisplayName','Hopf branches');
plot(nsbr1_pm(1,:), nsbr1_pm(2,:), 'Color',cm(2,:),...
     'DisplayName','Neimark-Sacker branches');
plot(nsbr2_pm(1,:), nsbr2_pm(2,:), 'Color',cm(2,:),...
     'HandleVisibility','off');
plot(nsbr3_pm(1,:), nsbr3_pm(2,:), 'Color',cm(2,:),...
     'HandleVisibility','off');
plot(lpc_br1_pm(1,:), lpc_br1_pm(2,:), 'Color',cm(3,:),...
     'DisplayName','Generalized Hopf');
plot(lpc_br2_pm(1,:), lpc_br2_pm(2,:), 'Color',cm(3,:),...
     'HandleVisibility','off');
% Add bifurcation points
plot(hoho.parameter(ind.zeta), hoho.parameter(ind.tau),...
     'k.', 'MarkerSize',8, 'DisplayName','Hopf Hopf point')
plot(hoho2.parameter(ind.zeta), hoho2.parameter(ind.tau),...
     'k.', 'MarkerSize',8, 'HandleVisibility','off')
plot(genhpts(1).parameter(ind.zeta),...
     genhpts(1).parameter(ind.tau), 'b.', 'MarkerSize',8,...
     'DisplayName','Generalized Hopf point')
plot(genhpts(2).parameter(ind.zeta),...
     genhpts(2).parameter(ind.tau), 'b.', 'MarkerSize',8,...
     'HandleVisibility','off')
plot(genhpts(3).parameter(ind.zeta),...
     genhpts(3).parameter(ind.tau), 'b.', 'MarkerSize',8,...
     'HandleVisibility','off')
title('Bifurcation diagram in  $(\zeta, \tau)$ ')
xlabel('\zeta$', 'Interpreter', 'LaTeX')
ylabel('\tau$', 'Interpreter', 'LaTeX')
axis([-0.3 0.4 5 16])
legend()

```

Figure S12 shows the resulting bifurcation diagram. There, we see that two generalized Hopf points are connected by the same limit point of cycles bifurcation curve.

S3.11 Simulation near Hopf-Hopf point with pydelay

In this last Section, we simulate the dynamics near the manually constructed Hopf-Hopf point at parameter values (109). As remarked before, the unfolding of the Hopf-Hopf point hoho is of ‘difficult’ type case VI. The normal form coefficients predict a stable invariant two-dimensional torus. Furthermore, this torus undergoes a bifurcation in which a three-dimensional torus is born. Since the DDE under investigation (108) does not contain any terms of order higher than three, the results for the normal form remain valid for the system. To confirm the unfolding in Figure 3, we fix the delay

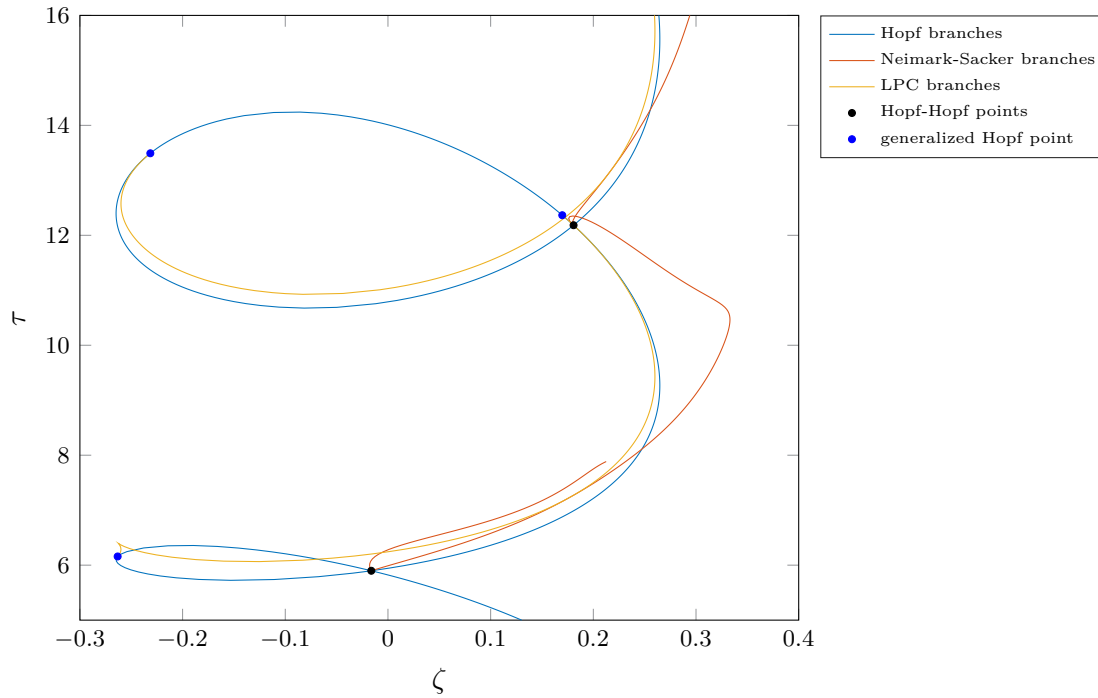


Figure S12: Unfolding from multiple codimension two points detected in the DDE (S4).

$\tau = 5.901783308978358$ and take for ζ consecutive the values

$$\begin{aligned}\zeta_1 &= -0.015485728828307, \\ \zeta_2 &= \zeta_1 - 0.0002, \\ \zeta_3 &= \zeta_1 - 0.0004, \\ \zeta_4 &= \zeta_1 - 0.000445, \\ \zeta_5 &= \zeta_1 - 0.0004496,\end{aligned}$$

see Figure S13a. The cross-sections in Figures S13b and S14d are generated with the following Python code using `pydelay`, see also the file `acs_simulation.py`

```
import numpy as np
import pylab as pl
from pydelay import dde23
from matplotlib import colors

# Number of time units
tfinal = 90000

# Define DDE
eqns = {
    'x': 'tau*y',
    'y': 'tau*(-x-0.1*x(t-1)-2*zeta*y' \
        '-0.52*y(t-1)+0.1*pow(x(t-1),3))'
}

# Set parameters
```

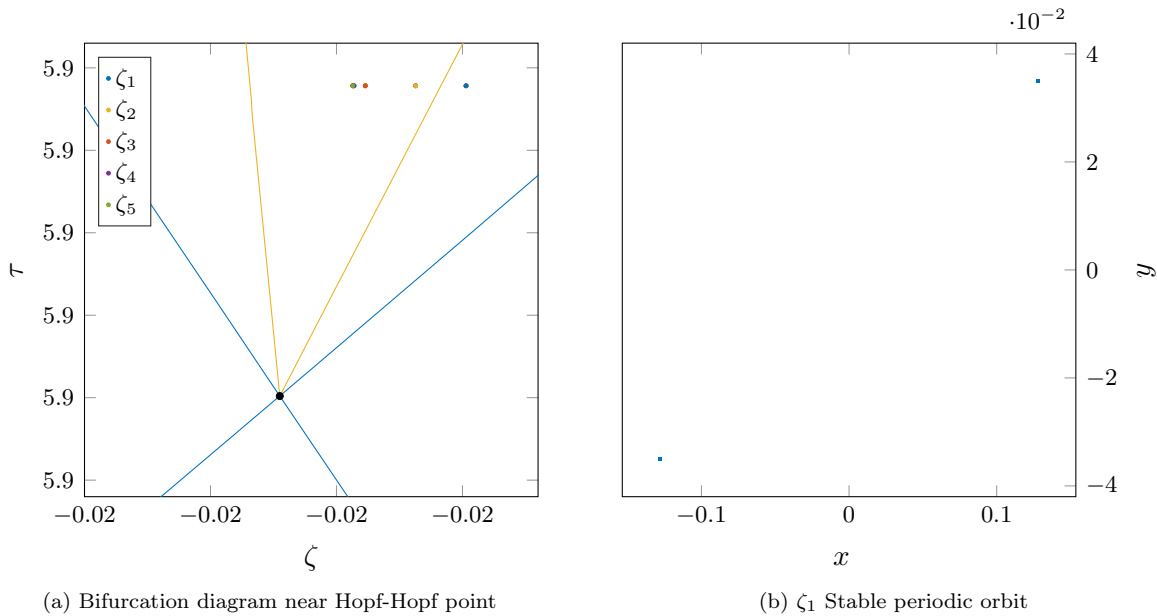


Figure S13: In (a), the points (ζ_i, τ) for $i = 1, \dots, 5$ of the parameter values where the simulation is performed are plotted. Note that ζ_4 and ζ_5 are almost indistinguishable. In (b), the Poincaré section of the simulation using `pydelay` at the parameters $(\zeta_2, \tau) = (-0.015685728828307, 5.901783308978358)$ shows a stable periodic orbit.

```

tau=5.901783308978358
zeta1=-0.015485728828307 # periodic orbit
zeta2=zeta1-0.0002      # torus
zeta3=zeta1-0.0004      # torus near bifurcation to 3d torus
zeta4=zeta1-0.000445    # 3d torus
zeta5=zeta1-0.0004496   # 3d torus near destruction

# Solve DDE
dde = dde23(eqns=eqns, params={'zeta':zeta2, 'tau':tau})
dde.set_sim_params(tfinal=tfinal, dtmax=0.1, AbsTol=1e-08,
                   RelTol=1e-06)
histfunc = {'x': lambda t: 0.01, 'y': lambda t: 0 }
dde.hist_from_funcs(histfunc, 51)
dde.run()

# Subtract solution components
a1=1;
dt=1e-04
sol = dde.sample(tfinal-tfinal/10, tfinal, dt)
soldelayed = dde.sample(tfinal-tfinal/10-a1, tfinal-a1, dt)
t = sol['t']
x = sol['x']
y = sol['y']
xdelayed = soldelayed['x']
zero_crossings = np.where(np.diff(np.sign(xdelayed)))[0]

# Scatter plot

```

```

x_cros=x[zero_crossings]
y_cros=y[zero_crossings]
params = {
    'figure.figsize': (15, 15),
    'axes.labelsize': 'x-large',
    'xtick.labelsize': 'x-large',
    'ytick.labelsize': 'x-large'
}
pl.rcParams.update(params)
fig = pl.figure()
pl.figure(1)
pl.scatter(x_cros,y_cros,s=0.8,color='royalblue')
pl.xlabel('$x$')
pl.ylabel('$y$')
pl.show()

```

In [Figure S13b](#) there are two dots, corresponding to a stable period orbit. Crossing the curve `ns1_br`, a stable two-dimensional torus branches off, see [Figure S14a](#). The torus still exists at $\zeta = \zeta_3$, as seen in [Figure S14b](#). Then, at $\zeta = \zeta_4$, only slightly smaller than ζ_3 , a three-dimensional torus is observed, see [Figure S14c](#). Lastly, [Figure S14d](#) shows the three-dimensional torus near the curve where the torus blows up.

S4 Transcritical-Hopf bifurcation in Van der Pol's oscillator with delayed position and velocity feedback

Consider the generalized van der Pol's oscillator with delayed feedback

$$\begin{cases} \dot{x}(t) = (\tau_0 + \mu_2) y(t), \\ \dot{y}(t) = (\tau_0 + \mu_2) [-x(t) - \varepsilon(x^2(t) - 1)y(t) + (1 + \mu_1)x(t - \tau) - 0.2y(t - 1) \\ \quad - 0.2x^2(t - 1) - 0.2x(t - \tau)y(t - 1) - 0.2y^2(t - 1) + 0.5x^3(t - 1)], \end{cases} \quad (\text{S5})$$

see [Section 8.4](#) and [\[3\]](#). The parameter $\varepsilon = 0.3$ is fixed. For $\tau_0 \approx 1.757290761249588$ a transcritical-Hopf bifurcation is located at $(\mu_1, \mu_2) = (0, 0)$.

Remark 26. This demonstration can be found in the directory `demos/tutorial/VII/vdpo` relative to the main directory of the DDE-BifTool package. Here, we omit the code to generate a system file. The system file `sym_vdpo_mf.m` has been generated with the script `gen_sym_vdpo.m`. Also, we assume that the DDE-BifTool package has been loaded as in [Listing S1](#). The code in [Sections S4.1](#) to [S4.8](#) highlights the important parts of the file `vdpo.m`.

S4.1 Set parameter names and funcs structure

We set the parameter names and define the `funcs` structure.

```

%% Set parameter names
parnames={'mu1','mu2','tau'};
cind=[parnames;num2cell(1:length(parnames))];
ind=struct(cind{:});
%% Set funcs structure
% We load the precalculated multilinear forms. These have been
% generated with the file gen_sym_vdpo.m.
funcs=set_symfuncs(@sym_vdpo_mf,'sys_tau',@( )ind.tau);

```

S4.2 Stability and normal form coefficients of the transcritical-Hopf point

We construct a steady-state at the transcritical-Hopf point and calculate its stability.

```
%% Construct transcritical-Hopf bifucation point
stst=dde_stst_create('x',[0;0],'parameter',[0 0 1]);
% Calculate stability
method=df_mthod(funcs,'stst');
stst.stability=p_stabil(funcs,stst,method.stability);
stst.stability.l1
```

The MATLAB console shows the following output.

```
ans =

    0.0000 + 0.0000i
   -0.0000 + 2.4539i
   -0.0000 - 2.4539i
```

We have a zero eigenvalue and a pair of purely imaginary eigenvalues. Furthermore, the remaining eigenvalues have negative real parts. Next, we calculate the normal form coefficients and the transformation to the center manifold with the function `nmfm_zeho`, which implements the coefficients as derived in Sections 6.2 and 6.4. For this we need to set the argument `free_pars` to the unfolding parameter (μ_1, μ_2) . These coefficients will be used to start the continuation of the various branches emanating from the transcritical-Hopf point.

```
%% Coefficients of the parameter dependent normal form
ht=p_tohopf(funcs,stst);
ht=p_tozeho(ht);
unfolding_pars=[ind.mu1, ind.mu2];
ht=nmfm_zeho(funcs,ht,'transcritical',1,'free_pars',unfolding_pars);
ht.nmfm
```

The MATLAB console shows the following output.

```
ans =

struct with fields:

    g200: 0.2121
    g110: -0.1337 + 0.2672i
    g011: 0.4241
    g300: 0.4935
    g111: 1.0243
    g210: -0.8178 - 0.4283i
    g021: -0.3302 - 0.1646i
         b: 0.2121
         c: 0.4241
         d: -0.1337 - 5.4430i
         e: -0.2435
         s: 0.0899
    theta: -0.6303
transcritical: 1
    h200: [1x1 struct]
    h011: [1x1 struct]
    h020: [1x1 struct]
    h110: [1x1 struct]
         K: [2x2 double]
```

```
omega1: 0.4644
omega2: 1.2768
```

The normal form coefficients are such that

$$g_{011} \operatorname{Re}(g_{110}) = 0.4241 \operatorname{Re}(-0.1337 + 0.2672i) < 0.$$

Therefore, there are two Neimark-Sacker bifurcation curves predicted, see [Section 5.4](#).

S4.3 Set bifurcation parameter range and step size bounds

Before continuing the various branches emanating from the transcritical-Hopf point, we create the variable `brpars` containing parameter bounds and maximal stepsizes.

```
% Set bifurcation parameter range and step size bounds
brpars={ 'max_bound',[ind.mu1 0.0139; ind.mu2 0.0094],...
         'min_bound',[ind.mu1 -0.0190; ind.mu2 -0.0069 ],...
         'max_step' ,[ind.mu1 1.0e-02; ind.mu2 1.0e-02]};
```

S4.4 Continuing Hopf, transcritical and Neimark-Sacker bifurcation curves

We use the function `C1branch_from_C2point` to continue the various branches emanating from the singularity.

```
%% Continue Neimark-Sacker curves emanating from
% the transcritical-Hopf point
[trfuncs,nsbr,suc]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','TorusBifurcation',...
    'step',1e-04,'plot',0,brpars{:});
assert(all(suc(:)>0))
ntrsteps=27; [nsbr(1),suc]=br_contn(trfuncs,nsbr(1),ntrsteps);
assert(suc>0)
ntrsteps=30; [nsbr(2),suc]=br_contn(trfuncs,nsbr(2),ntrsteps);
assert(suc>0)

%% Continue Hopf curves emanating from fold-Hopf point
[~,hbr,suc]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','hopf',brpars{:},'step',1e-05,'plot',0);
assert(all(suc(:)>0))
for i=2:-1:1
nop=1000; hbr(i)=br_contn(funcs,hbr(i),nop);
hbr(i)=br_rvers(hbr(i));
hbr(i)=br_contn(funcs,hbr(i),nop);
end

%% Continue transcritical curve emanating from fold-Hopf point
[~,tcbr]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','fold',brpars{1:4},...
    'step',linspace(-8.0e-03,8.0e-03,10),'plot',0);
```

S4.5 Detect special points on the Hopf branches

We continue with detecting bifurcations on the Hopf branches. The Hopf points on the branch `hbr_wbifs(i)` ($i = 1, 2$) will contain the normal form coefficients `L1` and `L2`. These will be used to visualize the criticality of the Hopf points (sub or super) in the bifurcation diagram.


```

%% Detect special points on the Hopf branches
for i=2:-1:1
    [hbr_wbifs(i),hopftests(i),hc2_indices,hc2_types]=...
    LocateSpecialPoints(funcs,hbr(i));
    al{i}=arrayfun(@(x)x.parameter(ind.mu1),hbr_wbifs(i).point);
    figure(i); clf;
    plot(al{i},hopftests(i).zeho(1,:),'.-',al{i},zeros(size(al{i})));
    xlabel('$\mu_1$', 'Interpreter', 'LaTeX');
    ylabel('First Lyapunov coefficient (L1)')
    title('Criticality along Hopf bifurcation curve')
end

```

S4.6 Predictors

For comparison in the bifurcation diagram we obtain predictors for the various branches by setting the argument `predictor` to 1 and `step` to a range of ε -values when calling the function `C1branch_from_C2point`.

```

%% Predictors for Neimark-Sacker, Hopf and transcritical curves
[trfuncs,nsbr_pred]=C1branch_from_C2point(funcs,ht,...
    unfolding_pars,'codim2','zeho','codim1','TorusBifurcation',...
    'step',linspace(1e-05,4e-02,20),'predictor',true);
[~,tcbr_pred]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','fold',...
    'step',linspace(-8.0e-03,8.0e-03,10),'predictor',true);
[~,hbrsub_pred]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','hopf',brpars{:},...
    'step',linspace(-1e-01,0,40),'predictor',1);
[~,hbrsup_pred]=C1branch_from_C2point(funcs,ht,unfolding_pars,...
    'codim2','zeho','codim1','hopf',brpars{:},...
    'step',linspace(0,1e-01,40),'predictor',1);
% Correct super- and subcritical hopf branches for the first curve
tempbr=hbrsub_pred(1);
hbrsub_pred(1)=hbrsuper_pred(1);
hbrsuper_pred(1)=tempbr;

```

S4.7 Bifurcation diagram

We plot the obtained curves and the predictors for the Neimark-Sacker, Hopf, and transcritical curves with the following code.

```

%% Plot comparing computed and predicted curves
figure(4); clf; hold on;
plot(ht.parameter(ind.mu1),ht.parameter(ind.mu2),'k.'...
    'MarkerSize',12)
tcbr_pm_pred = [getpars(tcbr_pred,ind.mu1); ...
    getpars(tcbr_pred,ind.mu2)];
plot(tcbr_pm_pred(1,:),tcbr_pm_pred(2,:),'.','Color',cm(5,:));
plot(tcbr_pm(1,:),tcbr_pm(2,:),'Color',cm(5,:));
for i=2:-1:1
    nsbr_pm_pred{i} = [getpars(nsbr_pred(i),ind.mu1); ...
        getpars(nsbr_pred(i),ind.mu2)];
    hbrsub_pm_pred{i} = [getpars(hbrsub_pred(i),ind.mu1); ...
        getpars(hbrsub_pred(i),ind.mu2)];

```

```

hbrsup_pm_pred{i} = [getpars(hbrsup_pred(i),ind.mu1); ...
    getpars(hbrsup_pred(i),ind.mu2)];
plot(nsbr_pm_pred{i}(1,:),nsbr_pm_pred{i}(2,:),...
    '.', 'Color',cm(3,:));
plot(hbrsub_pm_pred{i}(1,:),hbrsub_pm_pred{i}(2,:),...
    '.', 'Color',cm(2,:));
plot(hbrsup_pm_pred{i}(1,:),hbrsup_pm_pred{i}(2,:),...
    '.', 'Color',cm(1,:));
plot(nsbr_pm{i}(1,:),nsbr_pm{i}(2,:), 'Color',cm(3,:));
plot(hbrsup_pm{i}(1,:),hbrsup_pm{i}(2,:), 'Color',cm(2,:));
plot(hbrsub_pm{i}(1,:),hbrsub_pm{i}(2,:), 'Color',cm(1,:));
end
legend({'transcritical Hopf','transcritical predictor',...
'transcritical curve','Neimark-Sacker predictor',...
'subcritical Hopf predictor','supercritical Hopf predictor',...
'Neimark-Sacker branch','subcritical Hopf branch',...
'supercritical Hopf branch'})
title('Neimark-Sacker curve emanating from the transcritical-Hopf point')
axis([-0.0190 0.0139 -0.0069 0.0094])
xlabel('$\mu_1$', 'Interpreter', 'LaTeX')
ylabel('$\mu_2$', 'Interpreter', 'LaTeX')
text(-0.0129,0.0038, 'I'); text(-0.0093,0.007, 'II');
text(0.008,-0.0052, 'III'); text(0.0115,-0.003, 'IV');
legend('Location','NorthWest'); box on
% Reverse the stacking order of the graphics
chi=get(gca, 'Children'); set(gca, 'Children', flipud(chi));

```

Figure S15 shows the resulting bifurcation diagram.

S4.8 Plot comparing computed and predicted periodic orbits

Lastly, we create a plot to compare the computed and predicted periodic orbits.

```

% Plot comparing computed and predicted periodic orbits
figure(5); clf; hold on;
genpars=@(br,i)ones(2,length(...
    br.point(1).profile(1,:)).*br.point(i).parameter(ind.mu1);
for i=1:23
    plot3(genpars(nsbr(1),i),nsbr(1).point(i).profile(1,:),...
        nsbr(1).point(i).profile(2,:), 'Color',cm(1,:));
end
for i=1:12
    plot3(genpars(nsbr_pred(1),i),...
        nsbr_pred(1).point(i).profile(1,:),...
        nsbr_pred(1).point(i).profile(2,:), 'Color',cm(2,:));
end
for i=1:20
    plot3(genpars(nsbr(2),i),nsbr(2).point(i).profile(1,:),...
        nsbr(2).point(i).profile(2,:), 'Color',cm(1,:));
end
for i=1:12
    plot3(genpars(nsbr_pred(2),i),...
        nsbr_pred(2).point(i).profile(1,:),...
        nsbr_pred(2).point(i).profile(2,:), 'Color',cm(2,:));
end
title('Comparison between computed and predicted periodic orbits')

```

```
xlabel('\mu_1'); ylabel('x'); zlabel('y');
view(3)
```

The resulting plot is show in [Figure S16](#).

S4.9 Simulation near transcritical-Hopf point with pydelay

We simulate the dynamics in regions III and IV of [Figure S15](#). Since the critical normal form coefficients are such that

$$s = 1, \quad \theta < 0, \quad e < 0,$$

a stable cycle and stable torus should be present. The simulation in regions I and II have also been carried out, but have been omitted here. The following code can be found in the file `vdpo_simulation.py`.

```
import numpy as np
import pylab as pl
from pydelay import dde23
from matplotlib import colors

# Number of time units
tfinal = 30000

# Define DDE
eqns = {
    'x' : '(tau0+mu2)*y',
    'y' : '(tau0+mu2)*((1+mu1)*x(t-tau)-0.2*y(t-tau))'
        '-0.2*pow(x(t-tau),2)'
        '-0.2*x(t-tau)*y(t-tau)-0.2*pow(y(t-tau),2)'
        '-epsilon*(pow(x,2)-1)*y-x+0.5*pow(x,3))'
}

# Set parameters
# Period cycle
tau0=1.757290761249588
params1 = {
    'mu1':0.0049,
    'mu2':-0.0031,
    'tau0':tau0,
    'epsilon':0.3,
    'tau':1
}aH
/tru
# Torus
params2 = {
    'mu1':-0.006871405962603,
    'mu2':0.003871232826592+0.00001,
    'tau0':1.757290761249588,
    'epsilon':0.3,
    'tau':1
}

# Solve DDE
dde = dde23(eqns=eqns, params=params1)
dde.set_sim_params(tfinal=tfinal, dtmax=0.1, AbsTol=1e-8, RelTol=1e-6)
histfunc = {'x': lambda t: 0, 'y': lambda t: -0.2 }
dde.hist_from_funcs(histfunc, 51)
```

```

dde.run()

# Subtract solution components
M=600
sol1 = dde.sample(tfinal-M, tfinal, 0.1)
t = sol1['t']
x = sol1['x']
y = sol1['y']

# Plot the solution in phase-space
pl.plot(x, y)
pl.show()

# Plot the solution in phase-space
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
# Subtract delayed solution component
soltau = dde.sample(tfinal-M-tau0, tfinal-tau0, 0.1)
xtau = soltau['y']

# Poincaré section
def poincaresection(x, xtau, y, x_label, y_label, val):
    zero_cross = np.where(np.diff(np.sign(xtau-val)))
    plt.figure(1)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.plot(x[zero_cross], y[zero_cross], '.', c='royalblue')
    plt.show()
    return

x_label='$x(t)$'
y_label='$x(t-\tau)$'
poincaresection(x, xtau, y, x_label, y_label, -0.009)

```

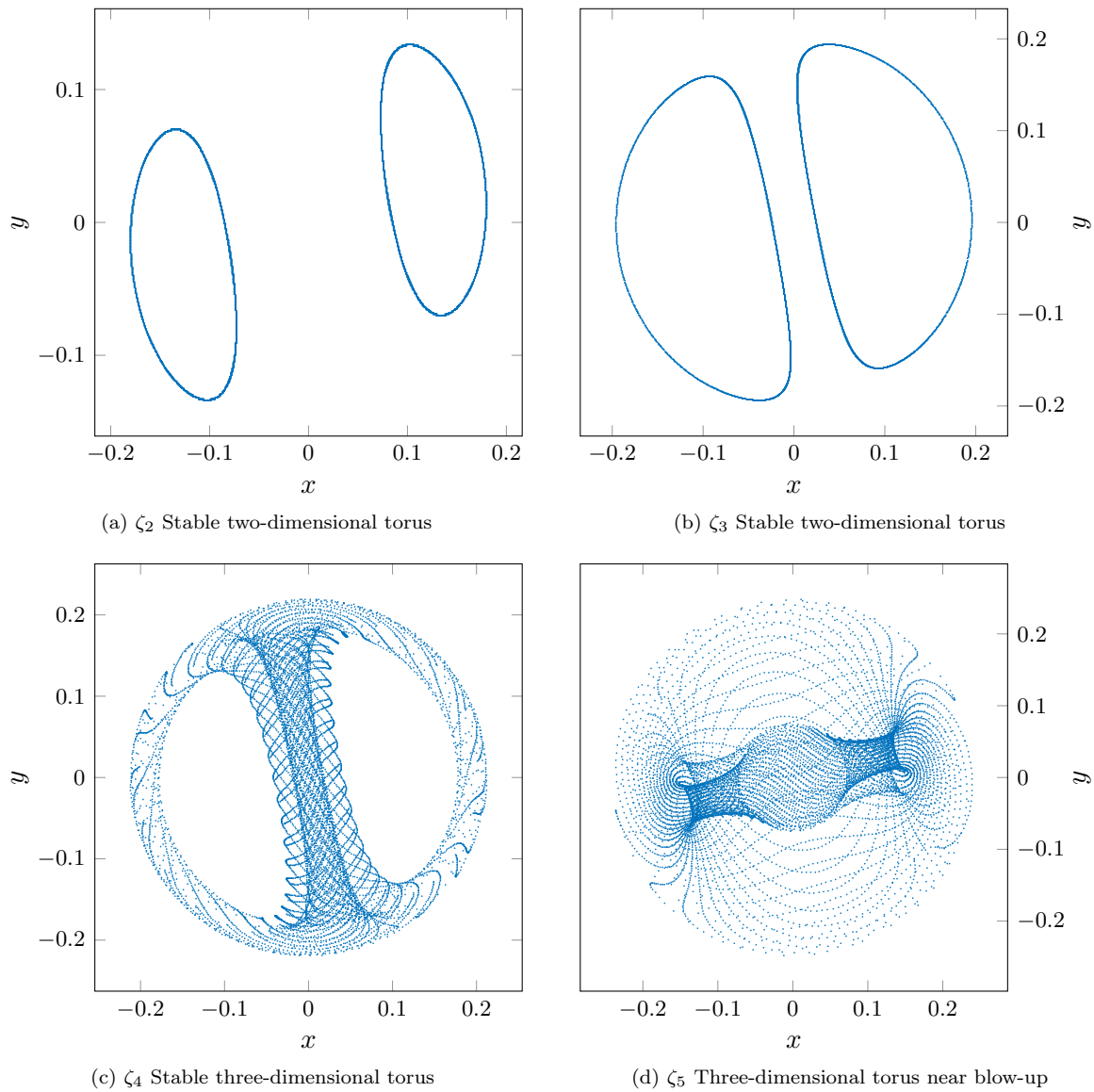


Figure S14: Simulation with `pydelay` illustrating the branching of a three-dimensional torus from a two-dimensional torus. We refer to the text for further description.

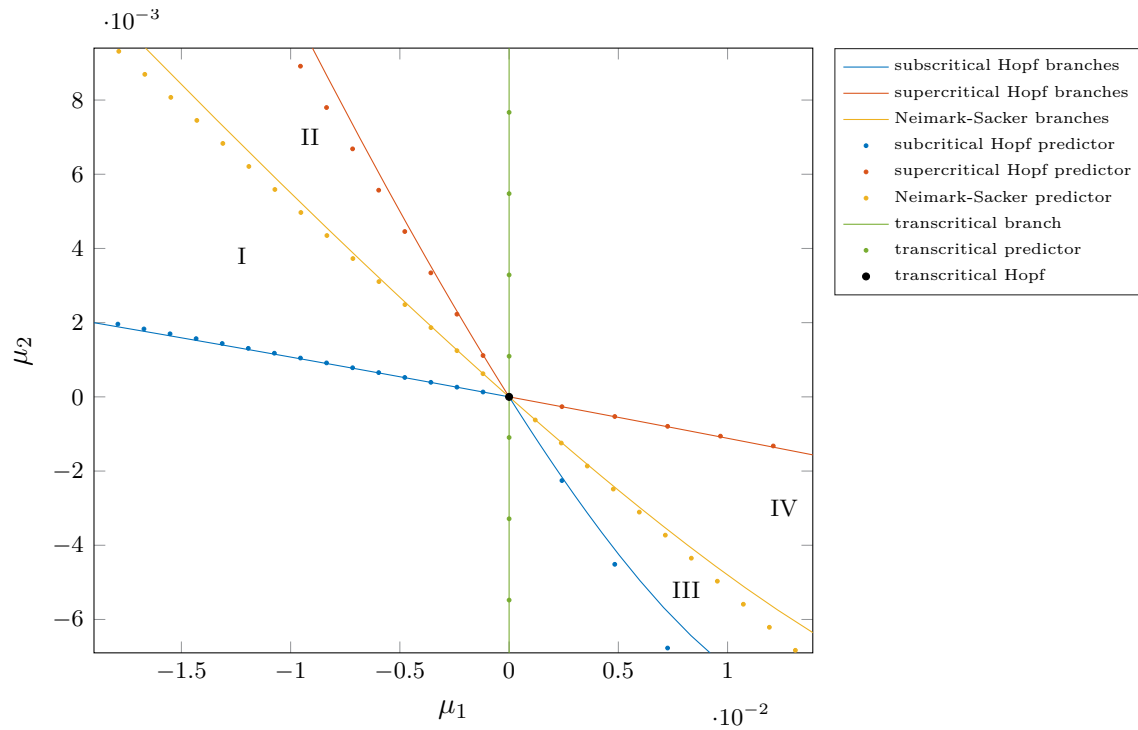


Figure S15: Bifurcation diagram near the transcritical-Hopf bifurcation in the delayed Van der Pol's oscillator given by (113). There are two supercritical Hopf curves (blue), two subcritical Hopf curves (red), two Neimark-Sacker curves (yellow), and one transcritical curve (green). We see that the predictors (dotted) give a good approximation for nearby values.

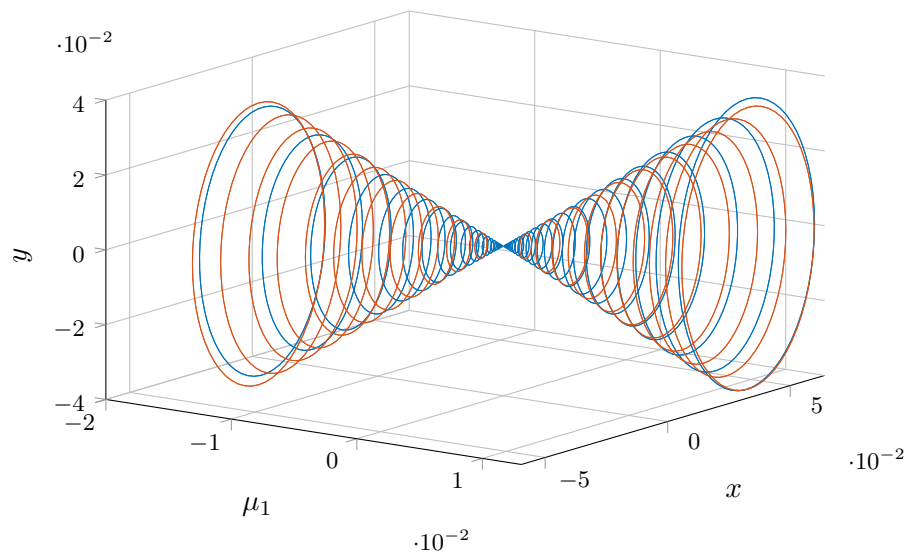


Figure S16: Comparison between predicted periodic orbits (red) and computed periodic orbits (blue) emanating from the transcritical-Hopf bifurcation

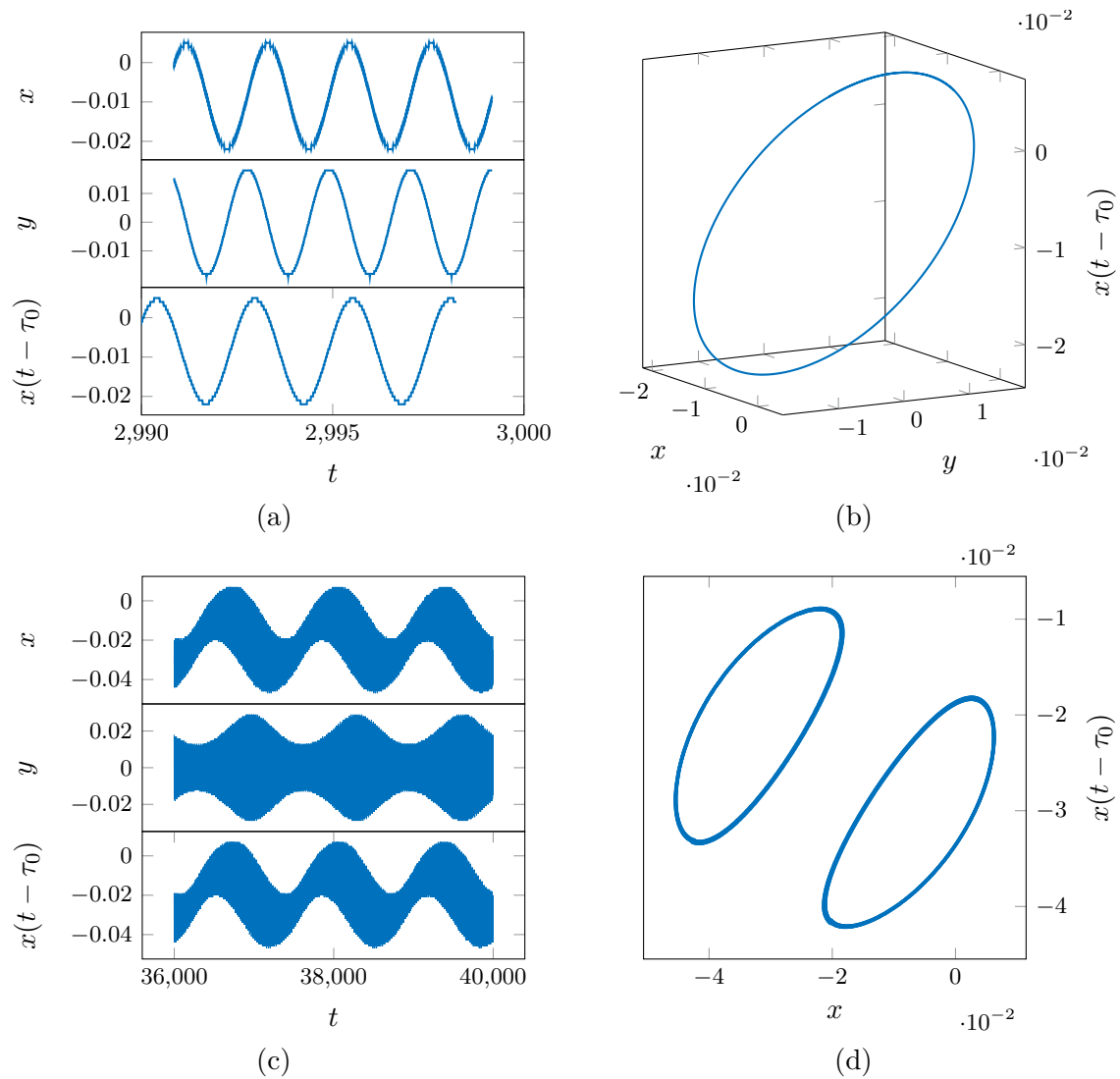


Figure S17: In (a) and (b), the periodic solution is shown to be present for parameter values in region III of [Figure S15](#). In (c), the torus present in IV of [Figure S15](#) is shown. In (d), a cross-section of the torus with $y(t) = -0.009$ in the phase-space $(x, x(t - \tau_0), y)$ is taken.

References for main text and supplement

- [1] R. AGRAWAL, D. JANA, R. K. UPADHYAY, AND V. S. H. RAO, *Complex dynamics of sexually reproductive generalist predator and gestation delay in a food chain model: double Hopf-bifurcation to chaos*, Journal of Applied Mathematics and Computing, (2016), pp. 1–35, doi:10.1007/s12190-016-1048-1.
- [2] W.-J. BEYN, A. CHAMPNEYS, E. DOEDEL, W. GOVAERTS, YU. A. KUZNETSOV, AND B. SANDSTEDE, *Numerical continuation, and computation of normal forms*, in Handbook of dynamical systems, Vol. 2, North-Holland, Amsterdam, 2002, pp. 149–219, doi:10.1016/S1874-575X(02)80025-X.
- [3] J. BRAMBURGER, B. DIONNE, AND V. G. LEBLANC, *Zero-Hopf bifurcation in the Van der Pol oscillator with delayed position and velocity feedback*, Nonlinear Dynam., 78 (2014), pp. 2959–2973, doi:10.1007/s11071-014-1638-0.
- [4] P. CLÉMENT, O. DIEKMANN, M. GYLLENBERG, H. J. A. M. HEIJMANS, AND H. R. THIEME, *Perturbation theory for dual semigroups. I. the sun-reflexive case*, Math. Ann., 277 (1987), pp. 709–725, doi:10.1007/BF01457866.
- [5] P. CLÉMENT, O. DIEKMANN, M. GYLLENBERG, H. J. A. M. HEIJMANS, AND H. R. THIEME, *Perturbation theory for dual semigroups. II. time-dependent perturbations in the sun-reflexive case*, Proc. Roy. Soc. Edinburgh Sect. A, 109 (1988), pp. 145–172, doi:10.1017/S0308210500026731.
- [6] P. CLÉMENT, O. DIEKMANN, M. GYLLENBERG, H. J. A. M. HEIJMANS, AND H. R. THIEME, *Perturbation theory for dual semigroups. III. nonlinear Lipschitz continuous perturbations in the sun-reflexive case*, in Volterra integrodifferential equations in Banach spaces and applications (Trento, 1987), vol. 190 of Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, 1989, pp. 67–89.
- [7] P. CLÉMENT, O. DIEKMANN, M. GYLLENBERG, H. J. A. M. HEIJMANS, AND H. R. THIEME, *Perturbation theory for dual semigroups. IV. the intertwining formula and the canonical pairing*, in Semigroup Theory and Applications (Trieste, 1987), vol. 116 of Lecture Notes in Pure and Appl. Math., Dekker, New York, 1989, pp. 95–116.
- [8] P. H. COULLET AND E. A. SPIEGEL, *Amplitude equations for systems with competing instabilities*, SIAM J. Appl. Math., 43 (1983), pp. 776–821, doi:10.1137/0143052.
- [9] V. DE WITTE, F. DELLA ROSSA, W. GOVAERTS, AND Y. A. KUZNETSOV, *Numerical periodic normalization for codim 2 bifurcations of limit cycles: computational formulas, numerical implementation, and examples*, SIAM J. Appl. Dyn. Syst., 12 (2013), pp. 722–788, doi:10.1137/120874904.
- [10] V. DE WITTE, W. GOVAERTS, Y. A. KUZNETSOV, AND H. G. E. MEIJER, *Analysis of bifurcations of limit cycles with Lyapunov exponents and numerical normal forms*, Phys. D, 269 (2014), pp. 126–141, doi:10.1016/j.physd.2013.12.002.
- [11] O. DIEKMANN, P. GETTO, AND M. GYLLENBERG, *Stability and bifurcation analysis of Volterra functional equations in the light of suns and stars*, SIAM Journal on Mathematical Analysis, 39 (2007), pp. 1023–1069, doi:10.1137/060659211.
- [12] O. DIEKMANN AND M. GYLLENBERG, *Abstract delay equations inspired by population dynamics*, in Functional Analysis and Evolution Equations, Birkhäuser, 2008, pp. 187–200, doi:10.1007/978-3-7643-7794-6_12.
- [13] O. DIEKMANN AND M. GYLLENBERG, *Equations with infinite delay: blending the abstract and the concrete*, J. Differential Equations, 252 (2012), pp. 819–851, doi:10.1016/j.jde.2011.09.038.

- [14] O. DIEKMANN, S. A. VAN GILS, S. M. VERDUYN LUNEL, AND H.-O. WALTHER, *Delay Equations: Functional-, Complex-, and Nonlinear Analysis*, Applied Mathematical Sciences, Springer, 1995, doi:10.1007/978-1-4612-4206-2.
- [15] K. DIJKSTRA, S. A. VAN GILS, S. G. JANSSENS, YU. A.. KUZNETSOV, AND S. VISSER, *Pitchfork-Hopf bifurcations in 1D neural field models with transmission delays*, Phys. D, 297 (2015), pp. 88–101, doi:10.1016/j.physd.2015.01.004.
- [16] Y. DING, J. CAO, AND W. JIANG, *Double Hopf bifurcation in active control system with delayed feedback: application to glue dosing processes for particleboard*, Nonlinear Dynam., 83 (2016), pp. 1567–1576, doi:10.1007/s11071-015-2431-4.
- [17] Y. DING AND W. JIANG, *Double Hopf bifurcation and chaos in Liu system with delayed feedback*, J. Appl. Anal. Comput., 1 (2011), pp. 325–349, doi:10.11948/2011023.
- [18] Y. DING, W. JIANG, AND P. YU, *Double Hopf bifurcation in a container crane model with delayed position feedback*, Appl. Math. Comput., 219 (2013), pp. 9270–9281, doi:10.1016/j.amc.2013.03.023.
- [19] K.-J. ENGEL AND R. NAGEL, *One-Parameter Semigroups for Linear Evolution Equations*, vol. 194 of Graduate Texts in Mathematics, Springer, New York, 2000, doi:10.1007/b97696.
- [20] K.-J. ENGEL AND R. NAGEL, *A Short Course on Operator Semigroups*, Universitext, Springer, 2006, doi:10.1007/0-387-36619-9.
- [21] K. ENGELBORGH, T. LUZYANINA, AND D. ROOSE, *Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL*, ACM Trans. Math. Software, 28 (2002), pp. 1–21, doi:10.1145/513001.513002.
- [22] T. FARIA AND L. T. MAGALHÃES, *Normal forms for retarded functional-differential equations and applications to Bogdanov-Takens singularity*, J. Differential Equations, 122 (1995), pp. 201–224, doi:10.1006/jdeq.1995.1145.
- [23] T. FARIA AND L. T. MAGALHÃES, *Normal forms for retarded functional differential equations with parameters and applications to Hopf bifurcation*, Journal of differential equations, 122 (1995), pp. 181–200, doi:10.1006/jdeq.1995.1144.
- [24] V. FLUNKERT AND E. SCHÖLL, *pydelay – a python tool for solving delay differential equations*, Nov. 2009, arXiv:0911.1633 [nlin.CD].
- [25] J. GE AND J. XU, *An efficient method for studying fold-Hopf bifurcation in delayed neural networks*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 21 (2011), pp. 1393–1406, doi:10.1142/S0218127411029100.
- [26] E. GEERARDYN AND N. SCHLÖMER, *matlab2tikz: a MATLAB/Octave script to convert native MATLAB/Octave figures to TikZ/Pgfplots figures*, 2008, <https://github.com/matlab2tikz/matlab2tikz> (accessed 2017/03/03). Version 1.1.0.
- [27] W. GOVAERTS, R. K. GHAZIANI, Y. A. KUZNETSOV, AND H. G. E. MEIJER, *Numerical methods for two-parameter local bifurcation analysis of maps*, SIAM J. Sci. Comput., 29 (2007), pp. 2644–2667, doi:10.1137/060653858.
- [28] W. J. F. GOVAERTS, *Numerical Methods for Bifurcations of Dynamical Equilibria*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000, doi:10.1137/1.9780898719543.

- [29] S. GUO, Y. CHEN, AND J. WU, *Two-parameter bifurcations in a network of two neurons with multiple delays*, *Journal of differential equations*, 244 (2008), pp. 444–486, [doi:10.1016/j.jde.2007.09.008](https://doi.org/10.1016/j.jde.2007.09.008).
- [30] S. GUO AND J. MAN, *Center manifolds theorem for parameterized delay differential equations with applications to zero singularities*, *Nonlinear Anal.*, 74 (2011), pp. 4418–4432, [doi:10.1016/j.na.2011.04.003](https://doi.org/10.1016/j.na.2011.04.003).
- [31] J. K. HALE, *Theory of Functional Differential Equations*, Springer, second ed., 1977, [doi:10.1007/978-1-4612-9892-2](https://doi.org/10.1007/978-1-4612-9892-2). Applied Mathematical Sciences, Vol. 3.
- [32] J. K. HALE AND S. M. VERDUYN LUNEL, *Introduction to Functional Differential Equations*, vol. 99 of Applied Mathematical Sciences, Springer, 1993, [doi:10.1007/978-1-4612-4342-7](https://doi.org/10.1007/978-1-4612-4342-7).
- [33] J. HINDMARSH AND R. ROSE, *A model of the nerve impulse using two first-order differential equations*, *Nature*, 296 (1982), pp. 162–164, [doi:10.1038/296162a0](https://doi.org/10.1038/296162a0).
- [34] J. L. HINDMARSH AND R. M. ROSE, *A model of neuronal bursting using three coupled first order differential equations*, *Proceedings of the Royal Society of London B: Biological Sciences*, 221 (1984), pp. 87–102, [doi:10.1098/rspb.1984.0024](https://doi.org/10.1098/rspb.1984.0024).
- [35] A.-V. ION, *An example of Bautin-type bifurcation in a delay differential equation*, *J. Math. Anal. Appl.*, 329 (2007), pp. 777–789, [doi:10.1016/j.jmaa.2006.06.083](https://doi.org/10.1016/j.jmaa.2006.06.083).
- [36] A.-V. ION AND R. M. GEORGESCU, *Bautin bifurcation in a delay differential equation modeling leukemia*, *Nonlinear Anal.*, 82 (2013), pp. 142–157, [doi:10.1016/j.na.2013.01.009](https://doi.org/10.1016/j.na.2013.01.009).
- [37] S. G. JANSSENS, *On a Normalization Technique for Codimension Two Bifurcations of Equilibria of Delay Differential Equations*, master’s thesis, Utrecht University, The Netherlands, 2010, <https://dspace.library.uu.nl/handle/1874/312252>. Corrections and updates are available via <http://sebastiaanjanssens.nl/pdf/normalization.pdf>.
- [38] S. G. JANSSENS, *A class of abstract delay differential equations in the light of suns and stars*, Jan. 2019, [arXiv:1901.11526](https://arxiv.org/abs/1901.11526) [math.DS].
- [39] H. JIANG, T. ZHANG, AND Y. SONG, *Delay-induced double Hopf bifurcations in a system of two delay-coupled van der Pol-Duffing oscillators*, *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 25 (2015), pp. 1550058, 18, [doi:10.1142/S0218127415500583](https://doi.org/10.1142/S0218127415500583).
- [40] W. JIANG AND H. WANG, *Hopf-transcritical bifurcation in retarded functional differential equations*, *Nonlinear Anal.*, 73 (2010), pp. 3626–3640, [doi:10.1016/j.na.2010.07.043](https://doi.org/10.1016/j.na.2010.07.043).
- [41] H. B. KELLER, *Lectures on numerical methods in bifurcation problems*, vol. 79 of Tata Institute of Fundamental Research Lectures on Mathematics and Physics, Published for the Tata Institute of Fundamental Research, Bombay; by Springer-Verlag, Berlin, 1987. With notes by A. K. Nandakumaran and Mythily Ramaswamy.
- [42] YU. A. KUZNETSOV, *Numerical normalization techniques for all codim 2 bifurcations of equilibria in ODEs*, *SIAM Journal on Numerical Analysis*, 36 (1999), pp. 1104–1124, [doi:10.1137/S0036142998335005](https://doi.org/10.1137/S0036142998335005).
- [43] YU. A. KUZNETSOV, *Elements of Applied Bifurcation Theory*, vol. 112 of Applied Mathematical Sciences, Springer, third ed., 2004, [doi:10.1007/978-1-4757-3978-7](https://doi.org/10.1007/978-1-4757-3978-7).
- [44] YU. A. KUZNETSOV, W. GOVAERTS, E. J. DOEDEL, AND A. DHOOGHE, *Numerical periodic normalization for codim 1 bifurcations of limit cycles*, *SIAM J. Numer. Anal.*, 43 (2005), pp. 1407–1435, [doi:10.1137/040611306](https://doi.org/10.1137/040611306).

- [45] YU. A. KUZNETSOV, H. G. E. MEIJER, B. AL HDAIBAT, AND W. GOVAERTS, *Improved homoclinic predictor for Bogdanov-Takens bifurcation*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 24 (2014), pp. 1450057, 12, [doi:10.1142/S0218127414500576](https://doi.org/10.1142/S0218127414500576).
- [46] YU. A. KUZNETSOV, H. G. E. MEIJER, W. GOVAERTS, AND B. SAUTOIS, *Switching to nonhyperbolic cycles from codim 2 bifurcations of equilibria in ODEs*, Phys. D, 237 (2008), pp. 3061–3068, [doi:10.1016/j.physd.2008.06.006](https://doi.org/10.1016/j.physd.2008.06.006).
- [47] M. LICHTNER, *Variation of constants formula for hyperbolic systems*, J. Appl. Anal., 15 (2009), pp. 79–100, [doi:10.1515/JAA.2009.79](https://doi.org/10.1515/JAA.2009.79).
- [48] S. MA AND Z. FENG, *Fold-Hopf bifurcations of the Rose-Hindmarsh model with time delay*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 21 (2011), pp. 437–452, [doi:10.1142/S0218127411028490](https://doi.org/10.1142/S0218127411028490).
- [49] H. G. E. MEIJER, F. DERCOLE, AND B. OLDEMAN, *Numerical bifurcation analysis*, in Mathematics of Complexity and Dynamical Systems. Vols. 1–3, Springer, 2012, pp. 1172–1194, [doi:10.1007/978-1-4614-1806-1_71](https://doi.org/10.1007/978-1-4614-1806-1_71).
- [50] J. PENG, L. WANG, Y. ZHAO, AND Y. ZHAO, *Bifurcation analysis in active control system with time delay feedback*, Appl. Math. Comput., 219 (2013), pp. 10073–10081, [doi:10.1016/j.amc.2013.04.014](https://doi.org/10.1016/j.amc.2013.04.014).
- [51] R. QESMI AND M. A. BABRAM, *Double Hopf bifurcation in delay differential equations*, Arab Journal of Mathematical Sciences, 20 (2014), pp. 280 – 301, [doi:10.1016/j.ajmsc.2013.10.002](https://doi.org/10.1016/j.ajmsc.2013.10.002).
- [52] A. F. RUSTON, *Fredholm Theory in Banach Spaces*, vol. 86 of Cambridge Tracts in Mathematics, Cambridge University Press, Cambridge, 1986, [doi:10.1017/CBO9780511569180](https://doi.org/10.1017/CBO9780511569180).
- [53] L. F. SHAMPINE AND S. THOMPSON, *Solving delay differential equations with dde23*, Appl. Numer. Math., (2001), pp. 441–458.
- [54] Z. SHEN AND C. ZHANG, *Double Hopf bifurcation of coupled dissipative Stuart-Landau oscillators with delay*, Appl. Math. Comput., 227 (2014), pp. 553–566, [doi:10.1016/j.amc.2013.11.044](https://doi.org/10.1016/j.amc.2013.11.044).
- [55] J. SIEBER, *Local bifurcations in differential equations with state-dependent delay*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 27 (2017), p. 114326, [doi:10.1063/1.5011747](https://doi.org/10.1063/1.5011747), [arXiv:1705.07550 \[math.DS\]](https://arxiv.org/abs/1705.07550).
- [56] J. SIEBER, K. ENGELBORGHES, T. LUZYANINA, G. SAMAËY, AND D. ROOSE, *DDE-BIFTOOL Manual - Bifurcation analysis of delay differential equations*, June 2014, [arXiv:1406.7144 \[math.DS\]](https://arxiv.org/abs/1406.7144).
- [57] Z. SONG AND J. XU, *Bursting near Bautin bifurcation in a neural network with delay coupling*, International Journal of Neural Systems, 19 (2009), pp. 359–373, [doi:10.1142/S0129065709002087](https://doi.org/10.1142/S0129065709002087). PMID: 19885964.
- [58] A. E. TAYLOR AND D. C. LAY, *Introduction to Functional Analysis*, John Wiley & Sons, New York, second ed., 1980.
- [59] S. A. VAN GILS, S. G. JANSSENS, YU. A. KUZNETSOV, AND S. VISSER, *On local bifurcations in neural field models with transmission delays*, Journal of Mathematical Biology, 66 (2013), pp. 837–887, [doi:10.1007/s00285-012-0598-6](https://doi.org/10.1007/s00285-012-0598-6), [arXiv:1209.2849 \[math.DS\]](https://arxiv.org/abs/1209.2849).
- [60] B. WAGE, *Normal form computations for delay differential equations in DDE-BIFTOOL*, master’s thesis, Utrecht University, The Netherlands, 2014, <https://dspace.library.uu.nl/handle/1874/296912>.

- [61] Y. WANG, H. WANG, AND W. JIANG, *Hopf-transcritical bifurcation in toxic phytoplankton-zooplankton model with delay*, J. Math. Anal. Appl., 415 (2014), pp. 574–594, [doi:10.1016/j.jmaa.2014.01.081](https://doi.org/10.1016/j.jmaa.2014.01.081).
- [62] X. P. WU, *Zero-Hopf bifurcation analysis of a Kaldor-Kalecki model of business cycle with delay*, Nonlinear Anal. Real World Appl., 13 (2012), pp. 736–754, [doi:10.1016/j.nonrwa.2011.08.013](https://doi.org/10.1016/j.nonrwa.2011.08.013).
- [63] J. XU AND K. W. CHUNG, *Effects of time delayed position feedback on a van der Pol-Duffing oscillator*, Phys. D, 180 (2003), pp. 17–39, [doi:10.1016/S0167-2789\(03\)00049-6](https://doi.org/10.1016/S0167-2789(03)00049-6).
- [64] Y. XU AND T. SHI, *Computation of double Hopf points for delay differential equations*, Open Math., 13 (2015), pp. 805–815, [doi:10.1515/math-2015-0076](https://doi.org/10.1515/math-2015-0076).
- [65] B. ZHEN AND J. XU, *Bautin bifurcation analysis for synchronous solution of a coupled FHN neural system with delay*, Commun. Nonlinear Sci. Numer. Simul., 15 (2010), pp. 442–458, [doi:10.1016/j.cnsns.2009.04.006](https://doi.org/10.1016/j.cnsns.2009.04.006).
- [66] B. ZHEN AND J. XU, *Fold-Hopf bifurcation analysis for a coupled FitzHugh-Nagumo neural system with time delay*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 20 (2010), pp. 3919–3934, [doi:10.1142/S0218127410028112](https://doi.org/10.1142/S0218127410028112).