Tractable nonlinear memory functions as a tool to capture and explain dynamical behaviors

Edgar Herrera-Delgado [*]
The Francis Crick Institute, 1 Midland Rd., London NW1 1AT, United Kingdom
and Department of Mathematics, King's College London, Strand, London WC2R 2LS, United Kingdom

James Briscoe [†]
The Francis Crick Institute, 1 Midland Rd., London NW1 1AT, United Kingdom

Peter Sollich [‡]
Department of Mathematics, King's College London, Strand, London WC2R 2LS, United Kingdom
and Institut für Theoretische Physik, University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

(Received 29 May 2020; accepted 4 September 2020; published 13 October 2020)

Mathematical approaches from dynamical systems theory are used in a range of fields. This includes biology where they are used to describe processes such as protein-protein interaction and gene regulatory networks. As such networks increase in size and complexity, detailed dynamical models become cumbersome, making them difficult to explore and decipher. This necessitates the application of simplifying and coarse graining techniques to derive explanatory insight. Here we demonstrate that Zwanzig-Mori projection methods can be used to arbitrarily reduce the dimensionality of dynamical networks while retaining their dynamical properties. We show that a systematic expansion around the quasi-steady-state approximation allows an explicit solution for memory functions without prior knowledge of the dynamics. The approach not only preserves the same steady states but also replicates the transients of the original system. The method correctly predicts the dynamics of multistable systems as well as networks producing sustained and damped oscillations. Applying the approach to a gene regulatory network from the vertebrate neural tube, a well-characterized developmental transcriptional network, identifies features of the regulatory network responsible for its characteristic transient behavior. Taken together, our analysis shows that this method is broadly applicable to multistable dynamical systems and offers a powerful and efficient approach for understanding their behavior.

DOI: 10.1103/PhysRevResearch.2.043069

I. INTRODUCTION

In complex dynamical systems, comprising multiple interacting components, it can be difficult to identify causal mechanisms and to dissect the function of parts of a system. Nonlinearities and feedback complicate intuitive understanding and these difficulties increase with the size and complexity of a system. Examples include biological processes such as protein-protein interaction and gene regulatory networks [1,2]. Mathematical models of these systems allow exploratory analysis and can provide insight but become less practical as system size grows. More importantly, the complexity can obscure the explanation for unexpected or emergent behaviors that originate in the dynamics of a system. For these reasons, a variety of approaches have been developed to reduce the complexity of models while preserving desired features of their behavior. An important class of tools are dimensionality-reduction techniques that coarse grain parts of a system [3–5].

The Zwanzig-Mori formalism provides an exact dimensionality reduction of a dynamical system based on a separation into an arbitrary subnetwork, the components of which are tracked explicitly, and a bulk containing the components that are replaced with memory functions [6–9]. These functions describe how the current subnetwork state feeds back, through the activity of molecular species in the bulk, to affect the subnetwork at a later time. This approach, specifically its nonlinear version, was originally developed for the dynamics of physical systems [10] but later generalized by Chorin *et al.* [11,12], with related uses also in optimal coarse graining [13]. A limitation, however, is that the memory functions are generally impossible to calculate in closed form [11,12]. Although approximate expressions can be derived in special cases [14–18], this restricts the applicability of the formalism. An alternative is to map the nonlinear system to a physical system consistent with the original; while this can be effective it does not necessarily simplify the problem [19,20].

[*] Present address: Institut Curie, 26 rue d'Ulm, 75248 Paris Cedex 05, France.

[†] james.briscoe@crick.ac.uk

[‡] peter.sollich@uni-goettingen.de

Another option is to expand the dynamical equations around a fixed point and derive memory functions from this approximation [21,22], but for multistable or oscillatory systems the memory functions obtained in this way do not capture all the qualitative behaviors.

To address this limitation, we develop a method, based on the formalism of Ref. [11], that allows the calculation of memory functions for generic dynamical systems without prior knowledge of the dynamics. We make one assumption: the bulk must not generate fixed points beyond those of the subnetwork, more specifically it must have a unique steady state for any subnetwork state, as assumed also in e.g. [18]. This is a natural condition: the subnetwork must be able to produce all fixed points itself, otherwise coarse graining cannot succeed. As the starting point for the dimensionality reduced dynamics we use the quasi-steady-state (QSS) approximation, where the bulk is always in steady state with the current subnetwork state as has been used in other contexts [23]. Memory functions are then constructed to correct the projected subnetwork state, by accounting for departures of the bulk from its steady state. Our main technical result is an explicit solution for the functions capturing these memory effects, derived in a systematic expansion around the QSS approximation. We demonstrate that the approach accurately predicts the dynamics of systems that produce multiple steady states and even sustained or damped oscillations. We also illustrate its use by applying it to a gene regulatory network from the embryonic vertebrate neural tube [24]. This is a transcriptional network of four interacting transcription factors with well described transient dynamics. We show how the memory functions generated by this approach provide insight into the features of the regulatory network that produce this transient behavior. Taken together, the analysis introduces a broadly applicable method for the investigation and analysis of complex dynamical systems.

II. MATHEMATICAL DERIVATION

A. Initial definitions

Following Ref. [11], we start from a system with degrees of freedom $\boldsymbol{x}$ evolving deterministically in time according to some nonlinear functions $\boldsymbol{R}$:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{R}(\boldsymbol{x}). \tag{1}$$

We define the flow $\boldsymbol{\phi}(\boldsymbol{x}, t)$ as the state the system reaches at time t if it starts in some initial state $\boldsymbol{x}$; this function thus obeys $\boldsymbol{\phi}(\boldsymbol{x}, 0) = \boldsymbol{x}$ and $\frac{\partial}{\partial t}\boldsymbol{\phi}(\boldsymbol{x}, t) = \boldsymbol{R}(\boldsymbol{\phi}(\boldsymbol{x}, t))$. We want to understand the dynamics of some chosen set of observables that we denote by the vector $\boldsymbol{A}$. Such observables are functions of the state of the system, which we write as $\boldsymbol{A}(\boldsymbol{x})$. By analogy with the definition of $\boldsymbol{\phi}$, the time-dependent observables are then taken as

$$\boldsymbol{A}(\boldsymbol{x}, t) = \boldsymbol{A}(\boldsymbol{\phi}(\boldsymbol{x}, t)), \tag{2}$$

so $\boldsymbol{A}(\boldsymbol{x}, t)$ gives the value of the observables at time t if the system was initially in state $\boldsymbol{x}$. The resulting time evolution of the observables can again be described by a differential equation

$$\frac{\partial}{\partial t}\boldsymbol{A} = L\boldsymbol{A}(\boldsymbol{x}), \qquad L = \sum_i R_i(\boldsymbol{x})\frac{\partial}{\partial x_i}, \tag{3}$$

with the Liouvillian L, a linear differential operator. The general setup above requires us to track the full $\boldsymbol{x}$ dependence of the chosen observables $\boldsymbol{A}(\boldsymbol{x}, t)$. To achieve a reduction in dimensionality, Chorin [11] assumes that the value of x is determined by $\boldsymbol{A}$ at least statistically, i.e., has some probability distribution that (only) depends on the current $\boldsymbol{A}$. Averages (expectations) over this distribution are written as $E[\cdot|\boldsymbol{A}]$, and the average evolution of $\boldsymbol{A}$ is governed by $\boldsymbol{v}(\boldsymbol{A}) = E[L\boldsymbol{A}(\cdot)|\boldsymbol{A}]$. Chorin [11,12] showed that the corrections to this in the actual time evolution take the form of a memory term and a so-called random force $\boldsymbol{r}$, giving the general form for the time evolution of $\boldsymbol{A}$ as

$$\frac{d}{dt}\boldsymbol{A} = \boldsymbol{v}(\boldsymbol{A}) + \int_0^t dt'\, \boldsymbol{M}(\boldsymbol{A}(t'), t - t') + \boldsymbol{r}. \tag{4}$$

The memory function $\boldsymbol{M}(\boldsymbol{A}(t'), t - t')$ depends on time difference $\tau = t - t'$ and—nonlinearly—on the past observable value $\boldsymbol{A}(t')$. Its evolution with τ is governed by the *deviations* of the drift from $\boldsymbol{v}(\boldsymbol{A})$; this evolution reads for a general observable $g(\boldsymbol{x}, \tau)$:

$$\frac{\partial}{\partial \tau}g(\boldsymbol{x}, \tau) = Lg(\boldsymbol{x}, \tau) - E[(Lg)(\cdot, \tau)|\boldsymbol{A}(\boldsymbol{x})]. \tag{5}$$

The memory function is obtained from the observable that measures exactly such fluctuations in the drift of the observables $\boldsymbol{A}$:

$$\boldsymbol{F}(\boldsymbol{x}) = (L\boldsymbol{A})(\boldsymbol{x}) - E[L\boldsymbol{A}(\cdot)|\boldsymbol{A}(\boldsymbol{x})]. \tag{6}$$

We define an $\boldsymbol{F}(\boldsymbol{x}, \tau)$ via Eq. (5) from the initial condition $\boldsymbol{F}(\boldsymbol{x}, 0) = \boldsymbol{F}(\boldsymbol{x})$, and the memory function is then given explicitly as $\boldsymbol{M}(\boldsymbol{A}, \tau) = E[L\boldsymbol{F}(\cdot, \tau)|\boldsymbol{A}]$. The random force itself is $\boldsymbol{r}(\boldsymbol{x}, t) = \boldsymbol{F}(\boldsymbol{x}, t)$ and has a vanishing average at all times, $E[\boldsymbol{r}(\cdot, t)|\boldsymbol{A}] = 0$ [11]. In our context, this term represents effects that come from the bulk starting *away* from QSS. When the bulk starts in QSS, which is what we assume in the following, it vanishes and so can be discarded. While in Ref. [11] steady-state dynamics are discussed, this is not required for the above formalism to be applicable.

B. Subnetwork dynamics

With the random force discarded as above, Eq. (4) is a closed equation for the time evolution of the observables $\boldsymbol{A}$ and so achieves the desired dimensionality reduction. However, the memory function cannot in general be calculated in any closed form. We now show that this *can* be done within a systematic approximation for subnetwork dynamics. By this, we mean that we consider as the observables $\boldsymbol{A} = \boldsymbol{x}^{\mathrm{s}}$ a subset of $\boldsymbol{x}$, e.g., the concentrations of molecular species in a subnetwork of a large gene regulatory network. We denote the degrees of freedom in the rest of the network, the bulk, by $\boldsymbol{x}^{\mathrm{b}}$ and write out the components of the general time evolution Eq. (1) as

$$\frac{d\boldsymbol{x}^{\mathrm{s}}}{dt} = \boldsymbol{R}^{\mathrm{s}}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}), \qquad \frac{d\boldsymbol{x}^{\mathrm{b}}}{dt} = \boldsymbol{R}^{\mathrm{b}}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}). \tag{7}$$

The Liouvillian then splits accordingly into

$$L = \sum_s R_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b})\frac{\partial}{\partial x_s} + \sum_b R_b(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b})\frac{\partial}{\partial x_b}, \tag{8}$$

with the sums running over subnetwork and bulk species, respectively. Here and below, subscripts always indicate individual species, while vectors with s and b superscripts collect all subnetwork and bulk quantities, respectively. With Eq. (8), the generic observable time evolution of Eq. (3) $(\partial/\partial t)\boldsymbol{x}^{\rm s} = L\boldsymbol{x}^{\rm s}$ reduces to Eqs. (7) as it should. We now need to choose how to define the expectation $E[\cdot|\boldsymbol{x}^{\rm s}]$. We do this so that, without the memory kernel, the reduced Eq. (4) corresponds to the simplification where the bulk dynamics equilibrates rapidly to any prevailing subnetwork state $\boldsymbol{x}^{\rm s}$, reaching a QSS value $\boldsymbol{x}^{\rm b*}$ defined by $d\boldsymbol{x}^{\rm b}/dt = 0$ or

$$\boldsymbol{R}^{\rm b}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s})) = 0. \tag{9}$$

As motivated in the Introduction, we will assume that this condition determines a unique bulk QSS $\boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s})$ for any $\boldsymbol{x}^{\rm s}$. The expectation required to construct the reduced Eq. (4) is taken accordingly as

$$E[g(\cdot)|\boldsymbol{x}^{\rm s}] = g(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s})), \tag{10}$$

i.e., by taking $\boldsymbol{x}^{\rm s}$ as prescribed and inserting for $\boldsymbol{x}^{\rm b}$ its QSS value. The average drift $\boldsymbol{v}(\boldsymbol{A}) = E[L\boldsymbol{A}(\cdot)|\boldsymbol{A}] = E[\boldsymbol{R}^{\rm s}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b})|\boldsymbol{x}^{\rm s}]$ now evaluates directly from Eq. (10) as

$$\boldsymbol{v}(\boldsymbol{x}^{\rm s}) = \boldsymbol{R}^{\rm s}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s})). \tag{11}$$

This is the QSS or fast bulk approximation to the subnetwork dynamics. Our main interest in the following lies in understanding the memory effects that account for the fact that the bulk is not in general fast, but evolves on a timescale comparable to that of the subnetwork. To determine the resulting memory function, we start from the definition of $\boldsymbol{F}(\boldsymbol{x})$, which from Eq. (6) has components

$$F_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) = R_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) - R_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s})). \tag{12}$$

The main challenge is now to calculate the evolution of this observable in time according to Eq. (5). This is not feasible in general but we can develop a systematic approximation by *linearizing* in deviations of the bulk degrees of freedom from the QSS, which we write as

$$F_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau) \approx \sum_b (x_b - x_b^*(\boldsymbol{x}^{\rm s}))f_{bs}(\boldsymbol{x}^{\rm s}, \tau). \tag{13}$$

The problem then reduces to finding the evolution of $f_{bs}(\boldsymbol{x}^{\rm s}, \tau)$ from the initial condition $f_{bs}(\boldsymbol{x}^{\rm s}, 0) \equiv f_{bs}^0(\boldsymbol{x}^{\rm s})$ obtained by linearizing Eq. (12),

$$f_{bs}^0(\boldsymbol{x}^{\rm s}) = \frac{\partial R_s}{\partial x_b}, \tag{14}$$

where the derivatives here and below are evaluated at $(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b*}(\boldsymbol{x}^{\rm s}))$ unless otherwise specified.

C. Memory evolution over time

To derive our memory function, we insert Eq. (13) into Eq. (5). Consistently applying the linearization as detailed in Appendix A yields the following equation for f_{bs}:

$$\frac{\partial}{\partial \tau} f_{bs} = \sum_{b'} l_{bb'} f_{b's} + \sum_{s'} R_{s'}\frac{\partial}{\partial x_{s'}} f_{bs} \tag{15}$$

with

$$l_{bb'} = J_{b'b} + \sum_{s'b''} (\boldsymbol{J}^{-1})_{b'b''}\frac{\partial R_{b''}}{\partial x_{s'}}\frac{\partial R_{s'}}{\partial x_b}, \tag{16}$$

where the Jacobian matrix $\boldsymbol{J}$ is defined as

$$J_{b''b'} = \frac{\partial R_{b''}}{\partial x_{b'}}. \tag{17}$$

The next step is to find a solution $f_{bs}(\boldsymbol{x}^{\rm s}, \tau)$ for the partial differential equation given in Eq. (15). This can be done using the method of characteristics as the equation is linear in $f_{bs}(\boldsymbol{x}^{\rm s}, \tau)$ and only involves first derivatives, and gives the closed form solution (see Appendix B)

$$f_{bs}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{b'} E_{bb'}(\tau) f_{b's}^0(\boldsymbol{\phi}_v(\boldsymbol{x}^{\rm s}, \tau)). \tag{18}$$

Here the $E_{bb'}$ are elements of the time-ordered matrix exponential $\boldsymbol{E}(\tau) = \exp[\int_0^\tau d\tau' \, \boldsymbol{l}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\rm s}, \tau'))]$ and the propagation in time is performed with the flow ϕ_v for the QSS drift $\boldsymbol{v}(\boldsymbol{x}^{\rm s})$.

D. Memory function

We can now finally determine the memory function on subnetwork species s, which from the general framework set out above is

$$M_s(\boldsymbol{x}^{\rm s}, \tau) = E[LF_s(\cdot, \tau)|\boldsymbol{x}^{\rm s}]. \tag{19}$$

We insert the expansion Eq. (13) here and obtain after some algebra (see Appendix A) our main result, a simple expression for the memory function,

$$M_s(\boldsymbol{x}^{\rm s}, \tau) = \sum_{b'} c_{b'}(\boldsymbol{x}^{\rm s}) f_{b's}(\boldsymbol{x}^{\rm s}, \tau), \tag{20}$$

where we have denoted

$$c_{b'}(\boldsymbol{x}^{\rm s}) = \sum_{s'}\sum_{b''} (\boldsymbol{J}^{-1})_{b'b''}\frac{\partial R_{b''}}{\partial x_{s'}} R_{s'}. \tag{21}$$

These functions can be thought of as prefactors to the memory term. The general projected time evolution equation now takes the form

$$\frac{d}{dt}x_s = v_s(\boldsymbol{x}^{\rm s}(t)) + \int_0^t dt' \, M_s(\boldsymbol{x}^{\rm s}(t'), t - t'). \tag{22}$$

The first term contains the QSS drift while the second one represents the memory correction to this, which is expressed in terms of the memory function Eq. (20). Our derivation allows this memory to cover the behavior around multiple fixed points of the system, due to its nonlinear dependence on $\boldsymbol{x}^{\rm s}$. The interpretation of our result Eq. (20) is that in a small time interval dt', $x_b - x_b^*$ will change by $c(\boldsymbol{x}^{\rm s}(t'))\,dt'$. This deviation from the QSS is propagated by the exponential matrix and affects the drift R_s at time t as captured by f_{bs}^0 in Eq. (18). In Appendix H, we compare Eq. (20) with the work of Ref. [18], which instead of the QSS assumption takes $\boldsymbol{x}^{\rm b} = 0$. This is unsuitable for the multistable systems we are interested in but we show that the method can be adapted to project to bulk QSS values (Appendix I). This leads to an expression similar to Eq. (20), but crucially without the propagation in time from t' to $t = t' + \tau$ (Appendix I).

E. Self-consistent approximation

Our linearization approach Eq. (13) implies that the memory term captures dynamical effects that are of first order in the deviations of the bulk network from its QSS. We will now develop an approximate self-consistent way of incorporating higher order corrections, which turns out also to simplify the numerical evaluation of the memory terms. Consider the factor $f^0_{bs}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau))$ that from Eqs. (18) and (20) appears in the memory function $M_s(\boldsymbol{x}^{\mathrm{s}}, \tau)$. In the actual memory integral this is evaluated for $\boldsymbol{x}^{\mathrm{s}}(t')$ and $\tau = t - t'$, i.e., as $f^0_{bs}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}(t'), t - t'))$. As explained above, ϕ_v is the flow generated only by the QSS drift, i.e., without memory corrections. But the memory terms change the flow, so we can make the approach self-consistent by substituting for ϕ_v the actual time evolution *with* memory. This corresponds to replacing

$$\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}(t'), t - t') \to \boldsymbol{x}^{\mathrm{s}}(t), \tag{23}$$

as we are just propagating the subnetwork state from $\boldsymbol{x}^{\mathrm{s}}(t')$ by a time difference $t - t'$ to $\boldsymbol{x}^{\mathrm{s}}(t)$. Making this replacement also in the matrix exponential in Eq. (18) changes the memory term $\mathcal{M}_s(t) = \int_0^t dt'\, M_s(\boldsymbol{x}^{\mathrm{s}}(t'), t - t')$ into

$$\tilde{\mathcal{M}}_s(t) = \sum_{b''} \int_0^t dt' \sum_{b'} c_{b'}(\boldsymbol{x}^{\mathrm{s}}(t'))\big(e^{\int_{t'}^t dt''\, \boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(t''))}\big)_{b'b''} \times f^0_{b''s}(\boldsymbol{x}^{\mathrm{s}}(t)). \tag{24}$$

The dependence on the subnetwork species s on which the memory acts is contained only in the—now t' independent—factor $f^0_{b'',s}(\boldsymbol{x}^{\mathrm{s}}(t))$. As shown in Appendix C, the memory integrals in the first line can then be calculated efficiently as solutions to differential equations, one for each bulk species b''. Conceptually, however, the self-consistent memory term is more complicated. In the original formulation Eq. (22), the memory is a superposition of separate effects from all past times t': the state $\boldsymbol{x}^{\mathrm{s}}(t')$ of the subnetwork affects the behavior of the bulk and feeds back into the subnetwork at time t. In Eq. (24), the way this feedback acts is additionally modulated by the entire time evolution of the subnetwork between times t' and t. In the applications considered below, both approaches yield similar quantitative results, hence which one to choose depends on the aim: for numerical calculations of memory effects, the self-consistent version is more efficient, whereas the memory functions themselves are easier to analyze in the original version because they depend—in addition to time difference, which always features—only on the subnetwork state at one time t'. Note that what we refer to as self-consistency is distinct from an approach widely used in the application of projection methods to physical systems (see, e.g., Refs. [25,26]), where equations of motion for correlation functions are considered and the relevant memory kernels are, via appropriate approximations, related back self-consistently to the correlation functions.

F. General memory properties

Both the ZMn and ZMs expressions for the memory term that we have derived are, as we have emphasized, nonlinear in $\boldsymbol{x}^{\mathrm{s}}$ and so not of the convolution form that would appear in linear ZM projection methods. This simpler form is recovered, however, in the dynamics near fixed points. To see this, we note that at a global fixed point $(\boldsymbol{x}^{\mathrm{s}*}, \boldsymbol{x}^{\mathrm{b}*})$, where $R_s = 0$, the last factor in the definition Eq. (21) of $c_{b'}$ vanishes. As c_b is a factor in both of our memory expressions, both receive zero contributions when $\boldsymbol{x}^{\mathrm{s}}(t')$ is at a fixed point. Assuming R_s and R_b are sufficiently smooth, we can therefore *linearize* the memory terms for dynamics near such a fixed point. In this linearization, all other factors in the memory are evaluated at the fixed point, and from this one deduces (see Appendix G) that the linearized forms of the ZMn and ZMs memory expressions are in fact identical. The memory terms then become time convolutions of $\boldsymbol{x}^{\mathrm{s}}(t') - \boldsymbol{x}^{\mathrm{s}*}$ with a memory kernel that depends only on time; Appendix G provides a numerical example. It follows from the general arguments in Appendix F that these kernels describe *exactly* the dynamics of the full linearized system, and so the equations with memory will correctly predict, e.g., the relaxation behavior near a stable fixed point.

G. Memory decomposition

In spite of their nonlinearity, it turns out to be possible to decompose our memory expressions into specific channels to analyze the contribution of interactions within a network. Building on the approach of Ref. [27], we take advantage of the two partial derivative expressions in Eqs. (14) and (21) to decompose the memory exactly into combinations of incoming and outgoing channels (Appendix D). The analogous construction for the self-consistent approximation is set out in Appendix E.

III. APPLICATIONS

To test the effectiveness of the method, we examine systems that contain multiple steady states, oscillatory behaviors, and complex transient dynamics. These are relevant in a wide range of physical and biological contexts.

A. Multistability

We first examine a series of multistable systems defined by mutually repressive Hill functions:

$$\frac{d}{dt}x_j = \frac{a}{1 + \sum_{i \neq j} x_i^n} - x_j. \tag{25}$$

The above equation constitutes an "or" logic because of the sum of the terms in the denominator, where even if only one repressor has a high concentration, the production rate will become very low. Such interactions lead to multistability in a wide variety of developmental systems [28].

We test the method on the simplest case with two nodes $\{x_1, x_2\}$, which leads to a system that cannot produce oscillations [29]. We place x_2 in the bulk and calculate the memory function for the single remaining subnetwork species x_1 [Fig. 1(a)]. This depends on the past concentration $x_1(t')$ and the time difference $\tau = t - t'$ [Fig. 1(b)]. We observe that

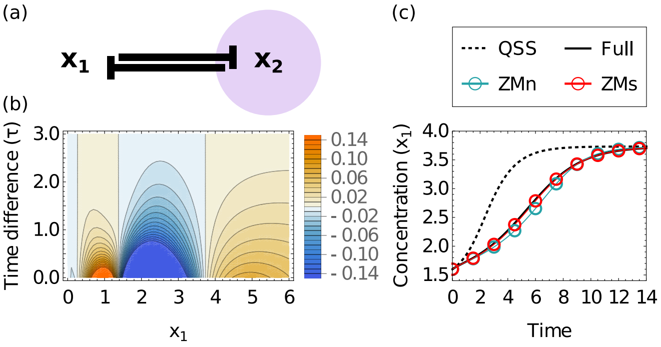

FIG. 1. (a) Network illustrating a bistable switch defined using cross-repressive Hill functions Eq. (25) with $a = 4$, $n = 2$ [steady states are $(x_1, x_2) = (6, 0.028)$ and $(0.028, 6)$ with an unstable fixed point at (1.46,1.46)]. For this and all other network illustrations, blunt arrows indicate repression; purple shading identifies the species placed in the bulk. (b) Memory function for the bistable switch shown as a function of (past) concentration x_1 of species 1 (x axis) and time difference τ (y axis). Memory function values range from negative to positive as indicated by the scale bar on the right and are capped by blue and orange outside the scale bar range. (c) Time course of the system demonstrates the capacity of both the nonlinear (ZMn, cyan) and self-consistent (ZMs, red) projections to capture the timescale and shape of transients of the full model (solid line) in reaching a stable fixed point. The QSS approximation (dashed line) significantly underestimates the length of the transient, showing that the ZM projections successfully correct for x_2 not being at QSS.

the memory becomes zero at each fixed point as expected from the discussion in Sec. II F above and from Ref. [27], where the memory was obtained as an expansion (to quadratic order) in deviations of $\boldsymbol{x}^{\mathrm{s}}$ from a fixed point. To leading order, the memory grows linearly with this deviation, and in line with this we see it changing sign at every fixed point. The sign of the memory in all cases is opposite to that of the drift, so the memory delays the relaxation time to the corresponding steady state. This makes intuitive sense as in the original system, the bulk species' state reacts relatively slowly to subnetwork changes, rather than infinitely fast as the QSS approximation assumes.

To test the accuracy of our method in capturing the transient temporal dynamics, we set the initial condition of x_1 to be close to the unstable fixed point of the QSS dynamics; here we are furthest from the stable fixed points and so can test the limits of the method. For the evaluation, we used as a baseline the full dynamics of the original system, setting x_2 at time zero to its QSS value with respect to the value of x_1. We compare this to the subnetwork dynamics predicted by the simple QSS dynamics, and by our approach, which includes memory corrections. For this example, we evaluate both the nonlinear memory description Eq. (22), ZMn, and the self-consistent memory Eq. (24), denoted ZMs below. We find that both replicate the behavior of the original system well, independently of whether the initial condition eventually leads to the low- or high-x_1 fixed point. The QSS approximation, on the other hand, reaches the steady state unrealistically fast [Fig. 1(c)]. Given that the ZMs method is substantially easier to implement for time course prediction (Appendix C), we concentrate on this approach below. Further justification for this comes from the fact that the self-consistent memory description is *exact* when R^{s} and R^{b} depend at most linearly on the bulk species, as we show in Appendix F. This exactness is not a trivial consequence of the fact that our approach is a linearization in $\boldsymbol{x}^{\mathrm{b}} - \boldsymbol{x}^{\mathrm{b}*}$, as it would otherwise hold also in the ZMn version. Biochemical systems with linear $\boldsymbol{x}^{\mathrm{b}}$ dependencies usually involve mass-action reactions and can produce bistable systems or oscillations [30,31], which we can then reproduce exactly with the ZMs projection (see Appendix F). A well-known example from physics is the Caldeira-Leggett model of a heat bath. This has a bulk composed of harmonic oscillators [32], and the resulting memory term was obtained exactly by Zwanzig [10,33]. Since in that system the bulk degrees of freedom appear linearly, our approach reproduces this solution but is more general in that it does not, for example, rely on the dynamics to be derived from a Hamiltonian.

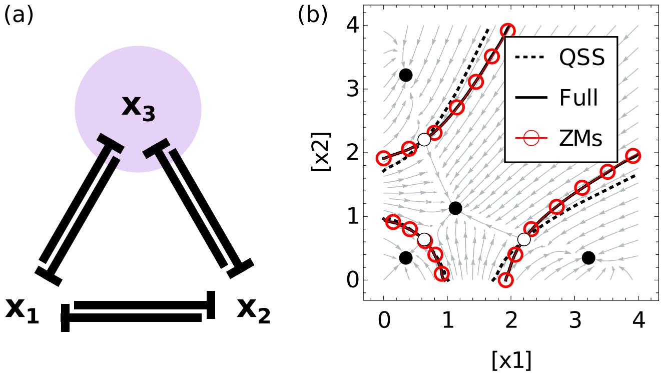

FIG. 2. (a) Network illustrating the cross-repressive tetrastable system Eq. (25); purple shading indicates the species placed in the bulk. Parameters are $a = 4$, $n = 2$. (b) Phase portrait indicating the basins of attraction of the four stable fixed points (black circles) and the unstable fixed points (white circles). The separatrices bounding each basin are shown for the full dynamics (solid lines), QSS approximation (dotted lines), and the subnetwork equations with memory (ZMs, red circles); stream plots are shown for the QSS approximation. The QSS approach shows a clear difference to the original system, while the boundaries set by ZMs and the full system are almost indistinguishable.

We next tested the approach on a tetrastable system defined in the same way as Eq. (25) with variables $\{x_1, x_2, x_3\}$. We consider the subnetwork containing x_1 and x_2, which will allow us to investigate the effect of the memory effects on the shapes of the basins of attraction of the different (stable) fixed points [Fig. 2(a)]. For the parameter values we use, there are four such fixed points for the full network: three where only one species has high concentration and the other two low, and one where all concentrations are equal [Fig. 2(b)]. The boundaries of the basins of attraction can be read as the points where, depending on its initial condition, the system chooses a different basin of attraction. In a biological setting, these choices could represent cell fate decisions, where a cell decides to adopt a specific specialized function and become a particular cell type (for a review, see Ref. [34]). We find that the QSS system fails to replicate the decision process of the original system, whereas the ZMs accurately identifies both the eventual steady state [Fig. 2(b)] *and* the timing to get to this state (data not shown).

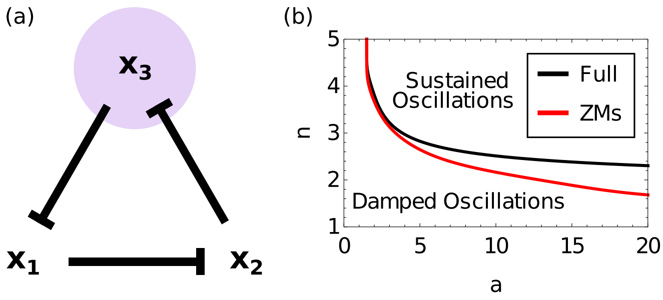

FIG. 3. (a) Network illustrating the repressilator system Eq. (26); purple shading indicates the species placed in the bulk. (b) Bifurcation diagram of the repressilator for system parameters a and n. The lines represent supercritical Hopf bifurcations. The QSS system can only produce damped oscillations, so has no bifurcation at all. The ZMs system (red line) shows a good qualitative match to the shape and position of the bifurcation of the full system (solid black line).

B. Oscillations

We further explore the ability of the subnetwork equations with memory to reproduce oscillations arising from a unidirectional repressive network. We use the repressilator system [35], which robustly generates oscillations due to delays arising from an odd number of nodes [29,36]. It has concentration variables $\{x_1, x_2, x_3\}$ and repressive interactions as shown in Fig. 3(a) and represented mathematically by

$$\frac{d}{dt}x_j = \frac{a}{1+x_{j-1}^n} - x_j, \tag{26}$$

where $x_0 \equiv x_3$. We first compare the bifurcation diagram that results from varying both system parameters a and n, in order to see whether the subnetwork equations with memory can replicate the 2D Hopf bifurcation of the original system, from damped to sustained oscillations [Fig. 3(b)].

In contrast to the QSS approximation, we find that the projection technique correctly replicates the existence of sustained oscillations and predicts a qualitatively correct bifurcation diagram. The period and amplitude of sustained oscilations in the relevant parameter regime is less well replicated (not shown). For damped oscillations, the subnetwork equations with memory work accurately in predicting the full temporal dynamics [Fig. 4(a)]. By contrast, the QSS approximation displays almost no oscillatory behavior and none of a sustained nature, highlighting the importance of memory effects for oscillatory transients.

To understand in more detail how memory generates oscillations, we analyze the corresponding memory functions. We first plot the memory amplitude, i.e., the value $M(\boldsymbol{x}^s, 0)$ for memory from the immediate past ($t' = t$) across the configuration space of our subnetwork [Fig. 4(b)] and observe two distinct regions with positive and negative memory amplitude, separated by a line where this amplitude vanishes (black). Plotting the time course from Fig. 4(a) in the same representation, we observe that it crosses the black line many times. The corresponding changes in the sign of the memory amplitude are what drives the oscillations seen in Fig. 4(a).

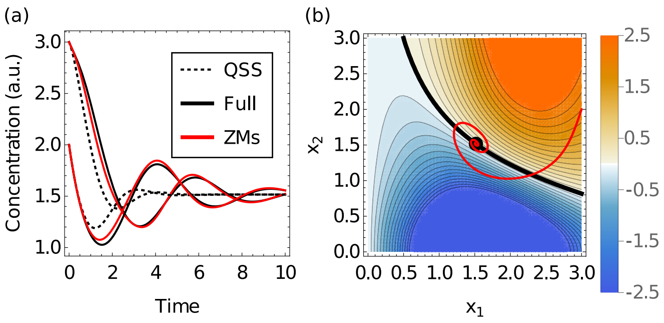

FIG. 4. (a) Damped oscillations generated by the repressilator [35] can be accurately reproduced by the projection approach (red line) while the QSS (dotted line) fails to replicate both the timing and the amplitude of the oscillations of the full system (solid black line). (b) Color map of memory amplitude (memory at $\tau = 0$) as a function of (x_1, x_2). The memory amplitude changes from negative (blue) to positive (orange) across the thick black line. Red curve: Parametric plot of the ZMs time course from the left; the memory repeatedly changes from negative to positive to drive the correct oscillations.

IV. NEURAL TUBE NETWORK (TRANSIENTS AND MULTISTABILITY)

Finally, we apply the projection approach to a biologically relevant system with several bifurcations and nontrivial dynamical properties, specifically the neural tube gene regulation network described in Ref. [24]. The network is sketched in Fig. 5(a) and its specific form and parametrization is given in Appendix J. In each cell of the neural tube, the network responds to the concentration of an extracellular signaling molecule, Shh. This molecule forms a gradient in the developing vertebrate neural tube which is interpreted by cells in the tissue through the actions of a Gene Regulatory Network (GRN) (for a recent review, see Ref. [37]). As a result of the concentration changing systematically along the neural tube,

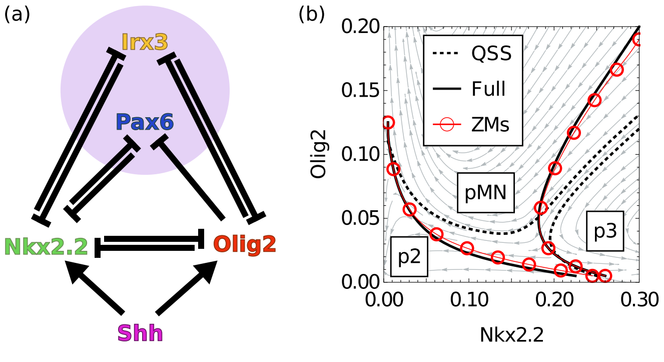

FIG. 5. (a) Neural tube network [24] defined by cross-repressive interactions between four transcription factors and activation by Sonic Hedgehog signaling (Shh), dependent on neural tube position; purple shading indicates choice of bulk. (b) Fate decision diagram for a neural tube position with three attractors (at position $p = 0.65$); solid lines indicate boundaries of basins of attraction that biologically separate different fate choices. Possible steady states are p3 (high Nkx2.2, right), pMN (high Olig2, top), and p2 (low Nkx2.2 and Olig2, and high Irx3 and Pax6, bottom left). Dashed lines indicate basin boundaries for the QSS approximation and red dots the basin boundaries for the ZMs projection; a stream plot is shown for the QSS system. The ZMs system very accurately reproduces the boundaries of the full system.

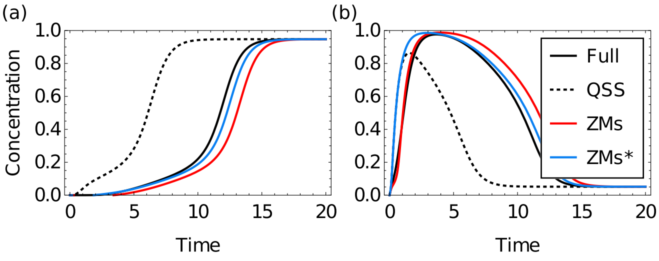

FIG. 6. Time courses of concentrations of Nkx2.2 (a) and Olig2 (b) in p3 domain ($p = 0.1$). A transient expression of Olig2 leading to a delay in Nkx2.2 expression is observed *in vivo*. The full system (solid line) and ZMs projections [with Nkx2.2 and Olig2 chosen as subnetwork, see Fig. 5(a), red and blue] qualitatively reproduce this behavior. In contrast, the QSS approximation (dashed line) is unable to capture the long Olig2 transient. ZMs∗ represents the removal of all memory functions except those specified in Fig. 7(a), which suffice to capture the observed transient.

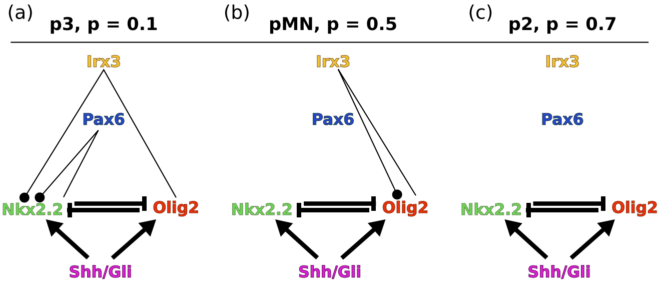

FIG. 7. Diagrams indicating the channels that have the largest memory contributions to the observed dynamics at three distinct neural tube positions. Black dots indicate the species receiving the memory contribution; the other end of each line is the species "sending" the memory. Contributions change according to the final steady state: p3 (a), pMN (b) and p2 (c).

we are in fact dealing here with a family of networks that vary with neural tube position, parametrized below in terms of p running from 0 to 1 (where p also represents the Shh input). The network contains four molecular species (called transcription factors in the gene regulation context), two of which generate a bistable switch by mutual cross-repression as in our first example above. Similarly to how we proceeded in Ref. [27], we therefore place these two molecular species (Nkx2.2 and Olig2) in the subnetwork, with the bulk provided by the two other species (Irx3 and Pax6).

We test the method at the position along the neural tube where the model has the most complexity, a region of tristability ($p = 0.65$), and compare with the original system and the QSS approximation [Fig. 5(b)]. We find that as for the tetrastable example above (Fig. 2), the projection accurately replicates the choice of steady state, in contrast to the QSS method [Fig. 5(b)]. With the memory included, three basins of attraction are predicted, of which one—labeled pMN—separates the other two (p2 and p3) so no direct transitions from p2 to p3 can occur. The QSS approximation is not just quantitatively inaccurate but loses this biologically important qualitative feature, predicting instead that the p2 and p3 basins border each other. At other neural tube positions, we also consistently find a good match between the original system and the ZMs projection approach (not shown).

We next analyze the temporal evolution of the systems at various neural tube positions, using the experimentally determined initial condition for Nkx2.2 and Olig2 (vanishing concentration); we again compare the ZMs description with the original system and the QSS reduction. The ZMs predictions show a good fit with the original system at all positions (Fig. 6 displays results for a position with a strongly nonmonotonic transient, $p = 0.1$), demonstrating that the memory functions are capable of accurately capturing not just final cell fate decisions but also the timing of such decisions. This temporal aspect is important for correct patterning in the neural tube as explored in Ref. [38].

A. Decomposing nonlinear memory functions

To understand how memory functions affect the patterning dynamics, we set out to understand their structure. We perform our self-consistent memory decomposition approach (Appendix E) and analyze the results to identify the memory channels with the most impact on the time courses based on their contribution along the trajectory (Appendix E). Performing this analysis for the different steady states, which along the neural tube form the so-called progenitor domains predicts the most important regulatory interactions contributing to the memory effect at each neural tube position (Fig. 7). This indicates marked differences in the most significant memory channels at different neural tube positions (see Appendix E).

To test the validity of our results we remove all channels identified as unimportant, setting them to QSS, thereby keeping only the channels shown in Fig. 7(a). Simulating the dynamics with only these memory functions results in dynamics that match closely those of the full simulation (Fig. 6), confirming the prediction that these channels dominate the memory effects.

We next investigated the experimentally validated transient in gene expression in the p3 domain (monostable with high Nkx2.2 in the steady state, $p = 0.1$, Fig. 6). Nkx2.2 induction is delayed in neural progenitor cells compared to Olig2 [39] and our analysis of the memory function provides insight into how this is achieved. The active memory channels ensure that Nkx2.2 is kept close to zero while Olig2 rises (Fig. 6). The dominant memory channels shown in Fig. 7(a) indicate that a different bulk species captures the history of each subnetwork species: Pax6 transmits the memory of Nkx2.2 and Irx3 the one of Olig2. The effect of these these bulk species is thus to delay Nkx2.2 expression based on the past expression of Nkx2.2 and Olig2.

Finally, we examined whether the effect of the two memory functions (one for Olig2 and one for Nkx2.2) that reflect the influence of the bulk is to increase the robustness of the system to initial conditions. For the system with memory, the delay in Nkx2.2 expression is present for multiple initial conditions, with trajectories crossing in a way that would be impossible to reproduce with a memoryless system (Fig. 8). From Fig. 7(a), the memory has two dominant channels reacting to changes in Nkx2.2 and Olig2, respectively. This ensures that if even one of the subnetwork species levels drifts away from zero, the memory pushes the path back into the "correct" direction. In the case of the memoryless system, the already short

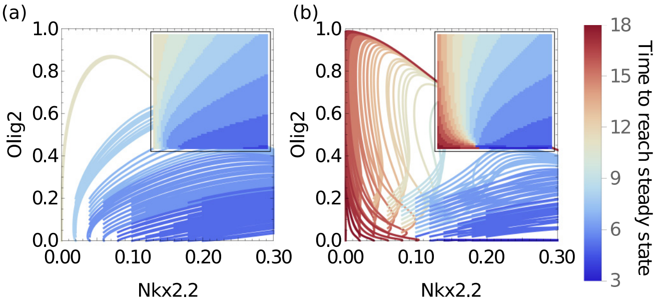

FIG. 8. Trajectories from multiple initial conditions starting near zero for Olig2 and Nkx2.2 in the p3 region ($p = 0.1$). The comparison emphasises the robustness provided by the memory functions [ZMs, (b)] in comparison to a memoryless system [QSS, (a)]. All trajectories reach the same attractor with high Nkx2.2 and low Olig2. The ZMs system behaves almost identically to the full system, with transient increases in Olig2 and large delays in reaching the steady state from a variety of initial conditions. Color indicates time taken to reach final steady state, quantified as Nkx2.2 deviating less than 1% from its asymptotic concentration. Insets have same axes as main plots and show time to final steady state from a given initial position.

transient observed in Fig. 6 disappears completely as soon as the initial conditions are no longer zero for Nkx2.2 [Fig. 8(a)]. In general, a transient is difficult to achieve in a 2D memoryless system where a very specific function would have to repress Nkx2.2 at both low and medium-high levels of Olig2. We find that the memory generated by the combination of Pax6 and Irx3 provides robustness to changes in the initial condition as the memory leads to low levels of Nkx2.2 during the initial phases of the transient [Fig. 8(b)].

V. DISCUSSION

We have developed a version of the Zwanzig-Mori formalism, building on the work of Chorin *et al.* [11], to obtain closed-form memory functions that can be used to reduce the dimensionality of a dynamical system far from equilibrium. Our method applies to general first-order nonlinear differential equations of arbitrary dimensionality. Systems with higher order time derivatives can in principle also be treated by introducing auxiliary variables for the lower order time derivatives in the conventional way, though we have not explored examples of this type. The method is constrained only in that the bulk must have a unique steady state given the subnetwork state, which we refer to as QSS (quasi-steady state). This restriction can be viewed as analogous to the assumption in the standard Zwanzig-Mori projection that, due to detailed balance, the bulk always reaches a Gibbs distribution for a given subnetwork state.

Similar to the method of averaging and other dimensionality reduction approaches [5,17,40], our method contains a timescale separation assumption as its baseline, where the bulk is assumed to reach its QSS arbitrarily quickly, but in contrast it then systematically finds the memory terms that correct for this assumption.

In our examples, the bulk parts of the network were usually chosen as small because, given the nonlinear cross-repressive nature of the biological interactions that we study, multiple bulk steady states might otherwise result. However, there is no general restriction that the bulk has to be small *per se* or smaller than the subnetwork, as long as the assumption of a unique bulk steady state is met. For example, in the neural tube network, one could choose a single subnetwork species (Nkx2.2) and three bulk species.

We have demonstrated the accuracy of the approach in capturing emergent dynamics such as nontrivial transients and sustained or damped oscillations. By construction, the method can capture multistability and we have shown its accuracy in predicting the basins of attraction that, in a biological context, delineate cell fate decisions. We subsequently applied the method to a biologically relevant system, the neural tube patterning network [24]. The reduced model captures the nontrivial dynamics of the original system through its nonmonotonic transients. In addition, it provides an understanding of the cause of such transients in gene expression and suggests that memory effects, stored in independent bulk nodes, provide robustness to initial conditions.

As more information is acquired about complex systems, methods like ours that coarse grain the elements of a system but conserve the dynamics will be crucial to provide an understanding of such systems. We have demonstrated the generality of this method and its flexibility and applicability to dynamical systems. In the biological context, the approach holds promise, e.g., for more complex networks that incorporate signaling and gene regulation dynamics, where it could be applied to distinguish the impact of these two effects onto the macroscopic behavior in, e.g., tissue patterning.

In the present paper, we have not explored the role of the random force, which captures effects of the bulk initially being away from QSS. However, our method gives a closed-form approximation for this term in the projected equations, which is simply the function F_s given by Eqs. (13) and (18). This should make it straightforward to explore random force effects in the future, e.g., to assess the importance of changes in bulk initial conditions in the spirit of our previous work [27].

A further interesting avenue of research would be the development of methods to capture the dynamics of systems even when parts of the network are unknown [19,20]. This approach could then be coupled with identifying the contribution of specific memory channels to the observed dynamics.

ACKNOWLEDGMENTS

We thank Matthias Fuchs and Alberto Pezzotta for helpful discussions and comments on the paper. This work was supported by the Francis Crick Institute, which receives its core funding from Cancer Research UK (FC001051), the UK Medical Research Council (FC001051), and Wellcome (FC001051); we also acknowledge funding from Wellcome [WT098325MA and WT098326MA]; and the European Research Council under European Union (EU) Horizon 2020 research and innovation program Grant No. 742138.

APPENDIX A: EXPANSION AROUND QSS

In this Appendix, we detail how we derive the memory evolution over time. We make use of our expansion Eq. (13) to find for the first term on the right hand side (RHS) of Eq. (5),

$$LF_s = \sum_b R_b(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) + \sum_{s'} R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) \sum_b (x_b - x_b^*) \frac{\partial}{\partial x_{s'}} f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) - \sum_{s'} R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) \sum_b \frac{\partial x_b^*}{\partial x_{s'}} f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) \tag{A1}$$

where the two last terms arise by differentiating the product $(x_b - x_b^*) f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau)$ w.r.t. $x_{s'}$ and we have not written the $\boldsymbol{x}^{\mathrm{s}}$ dependence of $\boldsymbol{x}^{\mathrm{b}*}$ for brevity. The second term on the RHS of Eq. (5) is the expectation of this, obtained by replacing $\boldsymbol{x}^{\mathrm{b}}$ by $\boldsymbol{x}^{\mathrm{b}*}$,

$$E[LF_s|\boldsymbol{x}^{\mathrm{s}}] = \sum_b R_b(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}*}) f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) - \sum_{s'} R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}*}) \sum_b \frac{\partial x_b^*}{\partial x_{s'}} f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau). \tag{A2}$$

Putting the two together gives for the time evolution Eq. (5) of F_s:

$$\begin{aligned} \frac{\partial F_s}{\partial \tau} &= \sum_b [R_b(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) - R_b(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}*})] f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) + \sum_{s'} R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) \sum_b (x_b - x_b^*) \frac{\partial}{\partial x_{s'}} f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau) \\ &\quad - \sum_{s'b} [R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}}) - R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}*})] \frac{\partial x_b^*}{\partial x_{s'}} f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau). \end{aligned} \tag{A3}$$

For consistency with Eq. (13), we now linearize the square brackets again in $\boldsymbol{x}^{\mathrm{b}} - \boldsymbol{x}^{\mathrm{b}*}$. In the second term, we similarly replace $R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}})$ by $R_{s'}(\boldsymbol{x}^{\mathrm{s}}, \boldsymbol{x}^{\mathrm{b}*})$ as the remaining factor in this term is already linear. Comparing then with the time derivative of the original linearized formula Eq. (13) gives, after appropriate relabelling of indices, the equation for the evolution of f_{bs} in time Eq. (15).

From the above expansion, one sees that the matrix $l_{bb'}$ in Eq. (15) takes the form

$$l_{bb'} = \frac{\partial R_{b'}}{\partial x_b} - \sum_{s'} \frac{\partial R_{s'}}{\partial x_b} \frac{\partial x_{b'}^*}{\partial x_{s'}}. \tag{A4}$$

The form Eq. (16) in the main text is obtained by using the identity

$$\frac{\partial x_{b'}^*}{\partial x_{s'}} = -\sum_{b''} (\boldsymbol{J}^{-1})_{b'b''} \frac{\partial R_{b''}}{\partial x_{s'}}. \tag{A5}$$

The latter can be obtained by differentiating Eq. (9) with respect to x_s.

To obtain the actual memory function from Eq. (19) is now straightforward as we have already worked out the required expectation in Eq. (A2). The first term on the r.h.s. of Eq. (A2), which we had previously kept to make the ensuing linearization easier to see, actually vanishes because of Eq. (9), yielding

$$M_s(\boldsymbol{x}^{\mathrm{s}}, \tau) = -\sum_{s'b'} R_{s'} \frac{\partial x_{b'}^*}{\partial x_{s'}} f_{b's}(\boldsymbol{x}^{\mathrm{s}}, \tau). \tag{A6}$$

Using again the identity Eq. (A5), we obtain our main result Eq. (20).

APPENDIX B: SOLUTION FOR F

In this Appendix, we find the solution $f_{bs}(\boldsymbol{x}^{\mathrm{s}}, \tau)$ to the differential Eq. (15). We start by restating the latter as

$$\frac{\partial}{\partial \tau} f_{bs} - \sum_{s'} v_{s'}(\boldsymbol{x}^{\mathrm{s}}) \frac{\partial}{\partial x_{s'}} f_{bs} = \sum_{b'} l_{bb'}(\boldsymbol{x}^{\mathrm{s}}) f_{b's}, \tag{B1}$$

where we have used that the factor $R_{s'}$ in Eq. (15) is just the effective drift $v_{s'}$ defined in Eq. (11). As the equation is linear in f_{bs}, its derivatives it can be solved using the method of characteristics (see, e.g., Ref. [41]). Calling the curve parameter for a characteristic u, the characteristic equations can be read off from Eq. (B1) as

$$\frac{d\tau}{du} = 1, \tag{B2}$$

$$\frac{dx_s}{du} = -v_s(\boldsymbol{x}^{\mathrm{s}}), \tag{B3}$$

$$\frac{df_{bs}}{du} = \sum_{b'} l_{bb'}(\boldsymbol{x}^{\mathrm{s}}) f_{b's}. \tag{B4}$$

Setting an arbitrary integration constant to zero, the first of these gives $\tau = u$. To solve Eq. (B3), we call $\boldsymbol{\phi}_v$ the flow generated by $\boldsymbol{v}(\boldsymbol{x}^{\mathrm{s}})$, which is defined as the solution of the differential equations

$$\frac{\partial}{\partial \tau} \boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau) = \boldsymbol{v}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau)), \qquad \boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, 0) = \boldsymbol{x}^{\mathrm{s}}. \tag{B5}$$

The solution of Eq. (B3) is then

$$\boldsymbol{x}^{\mathrm{s}}(u) = \boldsymbol{\phi}_v\big(\boldsymbol{x}_0^{\mathrm{s}}, -u\big), \tag{B6}$$

where $\boldsymbol{x}_0^{\mathrm{s}}$ is the value at the beginning of the characteristic curve ($u = 0$); the minus sign in the second argument of $\boldsymbol{\phi}_v$ reflects the backward in time propagation in Eq. (B3). We note for later that, as a consequence of Eq. (B6), the solution values at u_1 and u_2 are related by

$$\boldsymbol{x}^{\mathrm{s}}(u_2) = \boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}(u_1), -u_2 + u_1). \tag{B7}$$

Finally, the solution of Eq. (B4) is

$$f_{bs}(u) = \sum_{b'} \big(e^{\int_0^u du' \, \boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(u'))}\big)_{bb'} f_{b's}^0\big(\boldsymbol{x}_0^{\mathrm{s}}\big), \tag{B8}$$

using the initial condition Eq. (14) at $\tau = u = 0$. From Eq. (B4), we see that the matrix exponential appearing here must be time ordered, with earlier "times" u' appearing to the right of later ones.

It now remains to express $f_{bs}(u)$ in terms of $\boldsymbol{x}^{\mathrm{s}}(u)$ and $\tau(u) = u$. We fix a $\hat{u} = \hat{\tau}$ and call $\hat{\boldsymbol{x}}^{\mathrm{s}} = \boldsymbol{x}^{\mathrm{s}}(\hat{u})$. Using Eq. (B7) with $u_2 = u'$ and $u_1 = u$ then shows that the $\boldsymbol{x}^{\mathrm{s}}$-solution Eq. (B6) can be expressed in terms of $\hat{\boldsymbol{x}}^{\mathrm{s}}$ as

$$\boldsymbol{x}^{\mathrm{s}}(u') = \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}^{\mathrm{s}}, \hat{\tau} - u') \tag{B9}$$

and, in particular, $\boldsymbol{x}_0^{\mathrm{s}} = \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}^{\mathrm{s}}, \hat{\tau})$, so

$$f_{bs}(\hat{\boldsymbol{x}}^{\mathrm{s}}, \hat{\tau}) = \sum_{b'} \left(e^{\int_0^{\hat{\tau}} du'\, \boldsymbol{l}(\boldsymbol{\phi}_v(\hat{\boldsymbol{x}}^{\mathrm{s}}, \hat{\tau}-u'))}\right)_{bb'} f^0_{b's}(\boldsymbol{\phi}_v(\hat{\boldsymbol{x}}^{\mathrm{s}}, \hat{\tau})). \tag{B10}$$

Changing the integration variable to $\tau' = \hat{\tau} - u'$ and dropping the hats then gives the solution Eq. (18) announced in the main text. Note that as $\tau' = \hat{\tau} - u'$, the time ordering of the matrix exponential,

$$\boldsymbol{E}(\tau) = \exp\left(\int_0^{\tau} d\tau'\, \boldsymbol{l}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau'))\right), \tag{B11}$$

is such that the earlier τ' are now on the *left*. The appropriate time-ordered matrix exponential is defined formally via its Taylor series,

$$\boldsymbol{E}(\tau) = \mathbf{1} + \sum_{n=1}^{\infty} \int \prod_{i=1}^{n} d\tau_i\, \boldsymbol{l}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau_1)) \times \cdots \times \boldsymbol{l}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau_n)), \tag{B12}$$

with $\mathbf{1}$ the identity matrix and the integration in the other terms running over the range $0 < \tau_1 < \ldots < \tau_n < \tau$.

APPENDIX C: MAPPING OF SELF-CONSISTENT MEMORY TO DIFFERENTIAL EQUATIONS

We show in this Appendix how to map the subnetwork equations with self-consistent memory,

$$\frac{\partial}{\partial t} x_s = v_s(\boldsymbol{x}^{\mathrm{s}}(t)) + \tilde{\mathcal{M}}_s(t), \tag{C1}$$

to a set of differential equations. The self-consistent memory term $\tilde{\mathcal{M}}_s(t)$ is given by Eq. (24),

$$\tilde{\mathcal{M}}_s(t) = \sum_{b''} \int_0^t dt' \sum_{b'} c_{b'}(\boldsymbol{x}^{\mathrm{s}}(t'))\left(e^{\int_{t'}^{t} dt''\boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(t''))}\right)_{b'b''} f^0_{b''s}(\boldsymbol{x}^{\mathrm{s}}(t)), \tag{C2}$$

so can be written as

$$\tilde{\mathcal{M}}_s(t) = \sum_b m_b(t) f^0_{bs}(\boldsymbol{x}^{\mathrm{s}}(t)), \tag{C3}$$

with

$$m_b(t) = \int_0^t dt' \sum_{b'} c_{b'}(\boldsymbol{x}^{\mathrm{s}}(t'))\left(e^{\int_{t'}^{t} dt''\boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(t''))}\right)_{b'b}. \tag{C4}$$

It is then straightforward to check that

$$\frac{d}{dt} m_b(t) = c_b(\boldsymbol{x}^{\mathrm{s}}(t)) + \sum_{b'} m_{b'}(t) l_{b'b}(\boldsymbol{x}^{\mathrm{s}}(t)), \tag{C5}$$

where the second term arises from the t dependence of the matrix exponential. The $m_b(t)$ can therefore be obtained numerically by integrating the differential Eq. (C5) together with the subnetwork equations with (self-consistent) memory:

$$\frac{d}{dt} x_s(t) = v_s(\boldsymbol{x}^{\mathrm{s}}(t)) + \sum_{b'} m_{b'}(t) f^0_{sb'}(\boldsymbol{x}^{\mathrm{s}}(t)). \tag{C6}$$

The appropriate initial conditions for the auxiliary variables follow from Eq. (C4) as $m_b(0) = 0$.

APPENDIX D: CHANNEL DECOMPOSITION

We begin by writing the expression for the memory function explicitly, combining Eqs. (14), (18), (20), and (21):

$$\begin{aligned} M_s(\boldsymbol{x}^{\mathrm{s}}, \tau) &= \sum_{b's'b''} (\boldsymbol{J}^{-1})_{b'b''} \frac{\partial R_{b''}}{\partial x_{s'}} R_{s'} f_{b's}(\boldsymbol{x}^{\mathrm{s}}, \tau), && \text{(D1)} \\ &= \sum_{b's'b''} (\boldsymbol{J}^{-1})_{b'b''} \frac{\partial R_{b''}}{\partial x_{s'}} R_{s'} \sum_c E_{b'c}(\tau) f^0_{cs}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau)), && \text{(D2)} \\ &= \sum_{b's'b''} (\boldsymbol{J}^{-1})_{b'b''} \frac{\partial R_{b''}}{\partial x_{s'}} R_{s'} \sum_c E_{b'c}(\tau) \frac{\partial R_s}{\partial x_c}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau)), && \text{(D3)} \end{aligned}$$

where the first three factors are evaluated at $\boldsymbol{x}^{\mathrm{s}}$. We now swap index labels and group the sums into a more intuitive form:

$$M_s(\boldsymbol{x}^{\mathrm{s}}, \tau) = \sum_{s'} \sum_{bb'} \left(\sum_{b''} (\boldsymbol{J}^{-1})_{b''b'}(\boldsymbol{x}^{\mathrm{s}}) \frac{\partial R_{b'}}{\partial x_{s'}}(\boldsymbol{x}^{\mathrm{s}}) R_{s'}(\boldsymbol{x}^{\mathrm{s}}) E_{b''b}(\tau) \frac{\partial R_s}{\partial x_b}(\boldsymbol{\phi}_v(\boldsymbol{x}^{\mathrm{s}}, \tau)) \right). \tag{D4}$$

As discussed in the main text, the expression up to before the exponential represents a change in the deviation of the bulk species concentration $x_{b''}$ from its QSS values over some small time interval, in response to changes in the subnetwork concentrations $x_{s'}$ [see also Eq. (I4) below]. In the factor $\partial R_{b'}/\partial x_{s'}$, only those bulk species b' contribute whose time evolution depends explicitly on the subnetwork species s' driving the bulk time evolution via $R_{s'}$. The b' can then be interpreted as *outgoing channels* for the signal from s'. After propagation in the bulk network, the signal returns via another bulk species. Here only bulk species b contribute that appear explicitly in the time evolution of subnetwork species s as indicated by the factors $\partial R_s/\partial x_b$. The b can therefore be interpreted as *incoming channels*. Overall, we have memory effects from s' onto s, via an outgoing channel (s' to b') and an incoming channel (b to s). Consistent with this interpretation, the outgoing channel "susceptibilities" $\partial R_{b'}/\partial x_{s'}$ are evaluated for the past, i.e., sending, state $\boldsymbol{x}^{\mathrm{s}} \equiv \boldsymbol{x}^{\mathrm{s}}(t')$ of the subnetwork. The incoming channel susceptibilities $\partial R_s/\partial x_b$, on the other hand, are evaluated at the current time t as shown by the propagation via $\boldsymbol{\phi}_v$ across the time difference $\tau = t - t'$. Within the self-consistent approximation Eq. (C5), this propagation corresponds directly to evaluation at the current state $\boldsymbol{x}^{\mathrm{s}}(t)$.

APPENDIX E: SELF-CONSISTENT CHANNEL DECOMPOSITION

The channel decomposition of Sec. D can also be applied to the self-consistent memory approximation, as we now outline. Writing out the self-consistent memory term Eq. (C2) explicitly and reordering and relabelling terms as in Eq. (D4) gives

$$\begin{aligned}\tilde{\mathcal{M}}_s(t) &= \int_0^t dt' \sum_{s'}\sum_{bb'}\sum_{b''} \frac{\partial R_s}{\partial x_b}(\boldsymbol{x}^{\mathrm{s}}(t))\left(e^{\int_{t'}^{t} dt'' \boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(t''))}\right)_{b''b} \\ &\quad \times (\boldsymbol{J}^{-1})_{b''b'}(\boldsymbol{x}^{\mathrm{s}}(t'))\frac{\partial R_{b'}}{\partial x_{s'}}(\boldsymbol{x}^{\mathrm{s}}(t'))R_{s'}(\boldsymbol{x}^{\mathrm{s}}(t')) \\ &= \sum_{s'}\sum_{bb'} \frac{\partial R_s}{\partial x_b}(\boldsymbol{x}^{\mathrm{s}}(t))m_{sbb's'}(t), \qquad \text{(E1)}\end{aligned}$$

where

$$\begin{aligned}m_{sbb's'}(t) &= \int_0^t dt' \sum_{b''}(\boldsymbol{J}^{-1})_{b''b'}(\boldsymbol{x}^{\mathrm{s}}(t'))\frac{\partial R_{b'}}{\partial x_{s'}}(\boldsymbol{x}^{\mathrm{s}}(t'))R_{s'}(\boldsymbol{x}^{\mathrm{s}}(t')) \\ &\quad \times \left(e^{\int_{t'}^{t} dt'' \boldsymbol{l}(\boldsymbol{x}^{\mathrm{s}}(t''))}\right)_{b''b}. \qquad \text{(E2)}\end{aligned}$$

From this last representation, it follows that the $m_{sbb's'}(t)$ vanish at $t = 0$ and obey the differential equations:

$$\begin{aligned}\frac{d}{dt}m_{sbb's'}(t) &= (\boldsymbol{J}^{-1})_{bb'}(\boldsymbol{x}^{\mathrm{s}}(t))\frac{\partial R_{b'}}{\partial x_{s'}}(\boldsymbol{x}^{\mathrm{s}}(t))R_{s'}(\boldsymbol{x}^{\mathrm{s}}(t)) \\ &\quad + \sum_{b''} m_{sb''b's'}(t) l_{b''b}(\boldsymbol{x}^{\mathrm{s}}(t)). \qquad \text{(E3)}\end{aligned}$$

The channel-decomposed memory can therefore also be calculated from differential equations (see Appendix G). Of course, one only needs to find the $m_{sbb's'}$ for combinations (sb) and $(b's')$, where the corresponding channel susceptibilities are nonzero.

APPENDIX F: EXACTNESS OF MEMORY

We show that the self-consistent memory $m_b(t)$ is exact when both R_s and R_b contain at most linear terms in $\boldsymbol{x}^{\mathrm{b}}$. In such a case, the full system can be written as

$$R_s = v_s + \sum_{b'} \tilde{x}_{b'} f^0_{b's}, \qquad R_b = \sum_{b'} J_{bb'}\tilde{x}_{b'}, \qquad \text{(F1)}$$

where $\tilde{x}_b = x_b - x^*_b(\boldsymbol{x}^{\mathrm{s}})$ and the QSS value $\boldsymbol{x}^{\mathrm{b}*}(\boldsymbol{x}^{\mathrm{s}})$ is an arbitrary function of $\boldsymbol{x}^{\mathrm{s}}$. We now want to show that the $\tilde{x}_b$ correspond *exactly* to the m_b from the self-consistent ZMs method. To do this, we work out their evolution in time:

$$\begin{aligned}\frac{d}{dt}\tilde{x}_b &= R_b - \sum_{s'} R_{s'}\frac{\partial x^*_b}{\partial x_{s'}} \\ &= \sum_{b'} J_{bb'}\tilde{x}_{b'} + \sum_{s'}\left(v_{s'} + \sum_{b'}\tilde{x}_{b'}f^0_{b's'}\right)\sum_{b''}(\boldsymbol{J}^{-1})_{bb''}\frac{\partial R_{b''}}{\partial x_{s'}}. \\ &\qquad \text{(F2)}\end{aligned}$$

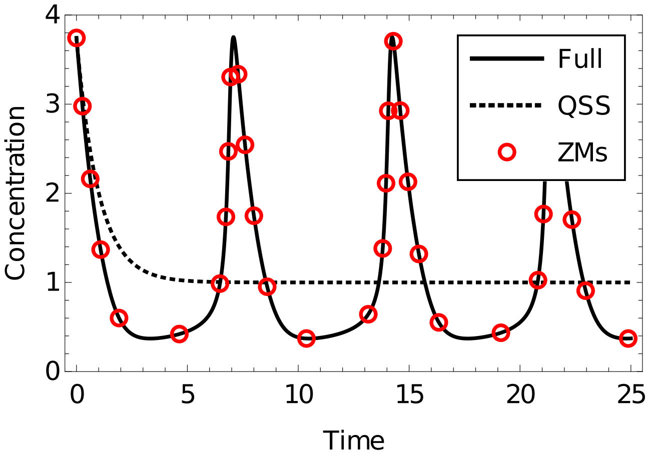

FIG. 9. Oscillating Brusselator system as described in Ref. [31]; we retain the concentration of the first species x_1 in the subnetwork and place the second species in the bulk. Parameters for the sustained oscillatory regime in this example are $A = 1$ and $B = 3$ in the notation of Ref. [31]. The trajectory of the self-consistent projection (red dots) captures that of the original system (solid line) exactly as expected from the general proof in Appendix F, while the QSS approximation (dotted line) fails qualitatively.

By using that for an $\boldsymbol{x}^{\mathrm{b}}$-linear system as assumed here, one has $f^0_{bs} = \partial R_s/\partial x_b$, the above can be rewritten as

$$\begin{aligned}\frac{d}{dt}\tilde{x}_b &= \sum_{b''}(\boldsymbol{J}^{-1})_{bb''}\frac{\partial R_{b''}}{\partial x_{s'}}v_{s'} \\ &\quad + \sum_{b'}\tilde{x}_{b'}\left(J_{bb'} + \sum_{s'b''}(\boldsymbol{J}^{-1})_{bb''}\frac{\partial R_{b''}}{\partial x_{s'}}\frac{\partial R_{s'}}{\partial x_{b'}}\right). \qquad \text{(F3)}\end{aligned}$$

Using then the definitions Eqs. (16) and (21), we obtain an expression equivalent to Eq. (C5),

$$\frac{d}{dt}\tilde{x}_b = c_b + \sum_{b'}\tilde{x}_{b'}l_{b'b}, \qquad \text{(F4)}$$

thus showing that $\tilde{x}_b = m_b$ when we start from the same initial condition $\tilde{x}_b = 0$, i.e., the bulk at QSS.

We test the above exactness statement on two different examples that have a linear dependence on a particular species but nonlinear dependencies on other species: a minimal bistable system as described in Ref. [30], and the Brusselator, which is capable of achieving limit cycles [31]. As expected from the above derivation, the self-consistent memory captures the behavior of both systems *exactly* (Figs. 9 and 10). As further shown in Fig 10, the original nonlinear projection method ZMn is also accurate at capturing the dynamics though not necessarily exact. The corresponding memory function is shown in Fig.11. (We note that the memory functions of the Brusselator grow exponentially in a way that forces memory terms to cancel out to zero at the fixed point; this leads to numerical challenges that we do not pursue here.)

APPENDIX G: LINEAR DYNAMICS

We discuss briefly the case of fully linear dynamics, where the dependence of R_s and R_b on *all* variables $\boldsymbol{x}^{\mathrm{s}}$ and $\boldsymbol{x}^{\mathrm{b}}$ (not

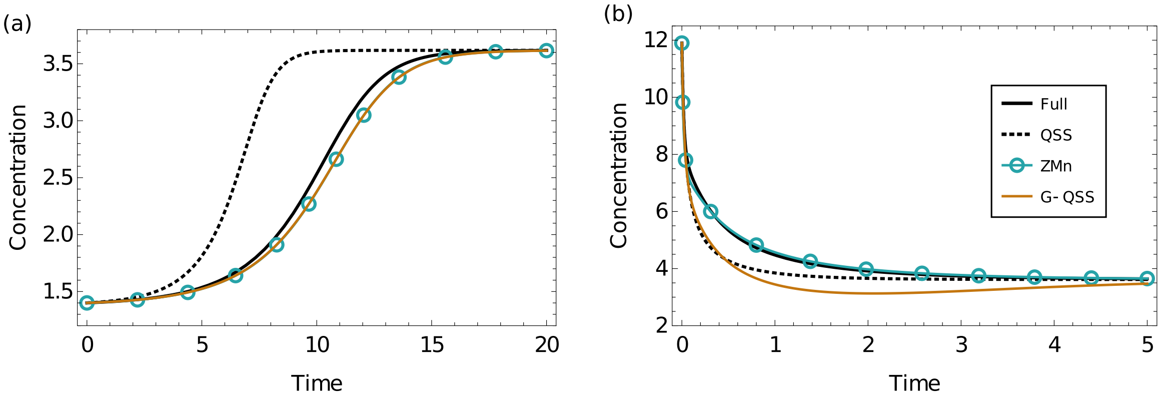

FIG. 10. Minimal bistable system with linear dependence on the second species as described in Ref. [30]; we choose x_1 for the subnetwork and place the second species in the bulk. Parameters for the bistable regime in this example are $k_1 = 10$, $k_2 = 1$, $k_3 = 2$, and $k_4 = 1$ [30]. The trajectory of the self-consistent ZMs projection again captures that of the original system (solid line) exactly (see Appendix F and Fig. 9) so is not plotted. G-QSS represents the Gouasmi *et al.* approach adapted to QSS projection (Appendix I). (a) $x_1(0) = 1.4$. At initial conditions near the fixed point, both the ZMn method (cyan dots) and G-QSS (orange line) behave similarly and accurately capture the full dynamics. (b) $x_1(0) = 11.9$. Further away from the final stable fixed point, the G-QSS predictions become increasingly inaccurate while the ZMn method continues to provide a good approximation.

just $\boldsymbol{x}^{\mathrm{b}}$ as in Appendix F) is only via constant and linear terms. Such a description can always be obtained by expanding linearly around a fixed point of the dynamics [21,27]. One then sees from Eqs. (16) and (14) that $l_{bb'}$ and f^0_{bs} are both constant, i.e., independent of $\boldsymbol{x}^{\mathrm{s}}$. Accordingly [compare Eqs. (18), (20), and (24)]. the memory functions of the ZMn and ZMs projections also become identical, and the corresponding channel decompositions are also the same.

To illustrate the linearized dynamics approach, we perform a channel decomposition of the amplitude (value at $\tau = 0$) of the linearized memory in the neural tube system as we did in Ref. [27], but now for the method derived in this study (Fig. 12). We find similar profiles to those found in Ref. [27]. The results highlight the relative weakness of the memory from Olig2 into Nkx2.2 via Pax6, supporting the conclusions of Ref. [27]. In addition, channel decomposition for the ZM trajectory shown in Fig. 6 provided memory terms affecting Nkx2.2 and Olig2 (Fig. 13). The method derived in this study is, however, significantly more powerful as it does not rely on an expansion near a steady state and gives access to the full memory and its channel decomposition as described in Appendixes D and E.

APPENDIX H: COMPARISON WITH ALTERNATIVE MEMORY FUNCTION APPROXIMATION

Gouasmi *et al.* [18] propose an approximation for the memory function for the case where the projection Eq. (10) is defined not by setting the bulk coordinates to their $\boldsymbol{x}^{\mathrm{s}}$-dependent QSS values, but simply to zero:

$$E[g(\cdot)|\boldsymbol{x}^{\mathrm{s}}] = g(\boldsymbol{x}^{\mathrm{s}}, 0). \tag{H1}$$

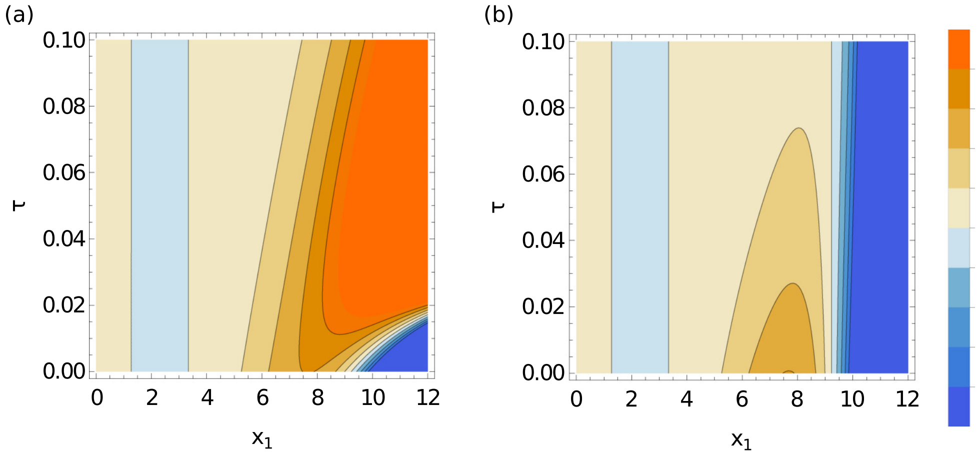

FIG. 11. Memory functions for the system detailed in Ref. [30] with the parameters chosen for Fig. 10. (a) Using the ZMn. (b) Using the method from Gouasmi *et al.* as extended to QSS projection in Appendix I. The x axis shows the concentration of the subnetwork species x_1 while the y axis indicates time difference τ. By construction, the two memory function approximations predict the same value (scale bar to the right) at $\tau = 0$, as they only differ in how they propagate the memory over time. At $\tau > 0$, the memory functions are relatively similar for $x_1 \in [0, 6]$ but become progressively different as x_1 grows beyond this range; for $x_1 \geqslant 10$, the G-QSS method predicts a negative memory function for all τ that leads to its poor performance as observed in Fig. 10.

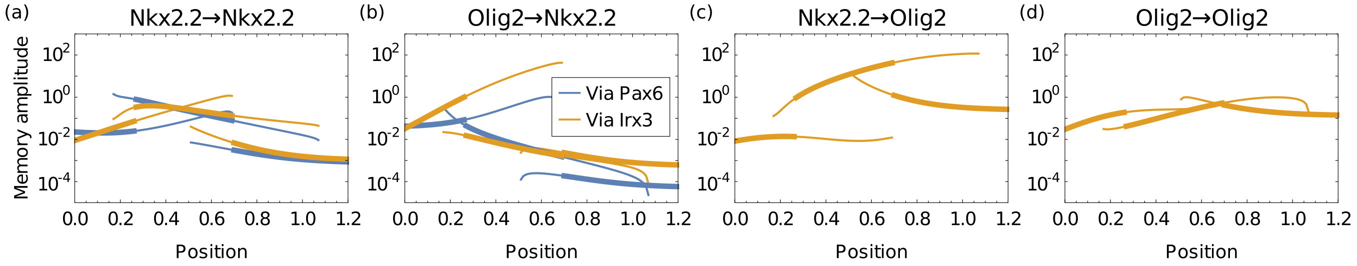

FIG. 12. Amplitudes (value at time difference $\tau = 0$) of nonlinear memory functions linearized around fixed points, for comparison with the approach of Ref. [27], where memory functions expanded around fixed points were calculated directly. (a) Amplitude of memory of Nkx2.2 to itself along the neural tube. There are multiple lines as the analysis was performed at all possible stable steady states. The vertical axis is logarithmic to make the range of amplitudes easier to appreciate. Colors identify the memory amplitude contribution from the two possible bulk channels, via Irx3 and Pax6, respectively. Thick lines indicate physiological states, while thin lines indicate states that are not usually observed in vivo. (b) Amplitude of memory of Nkx2.2 to levels of Olig2, shown along the neural tube. The memory via Pax6 is for the most part below the memory via Irx3, in each pair of corresponding curves. (c), (d) Amplitudes of memory of Olig2 to past Nkx2.2 (c) and to itself (d). No channel decomposition is performed as Olig2 receives memory only via the Irx3 channel.

The function $F_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau)$ still evolves according to Eq. (5), which written out now reads

$$\begin{aligned}\frac{\partial}{\partial\tau}F_s &= LF_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau) - E[LF_s(\cdot, \tau)|\boldsymbol{x}^{\rm s}], \qquad \text{(H2)}\\ &= \sum_{s'} R_{s'}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b})\frac{\partial F_s}{\partial x_{s'}} + \sum_b R_b(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b})\frac{\partial F_s}{\partial x_b},\\ &\quad - \sum_{s'} R_{s'}(\boldsymbol{x}^{\rm s}, 0)\frac{\partial F_s}{\partial x_{s'}}(\boldsymbol{x}^{\rm s}, 0, \tau)\\ &\quad - \sum_b R_b(\boldsymbol{x}^{\rm s}, 0)\frac{\partial F_s}{\partial x_b}(\boldsymbol{x}^{\rm s}, 0, \tau), \qquad \text{(H3)}\end{aligned}$$

where the very last factor is the x_b derivative of F evaluated at $\boldsymbol{x}^{\rm b} = 0$. The approximation of Ref. [18] amounts to ignoring the fact that the derivatives of F_s are evaluated at a different point in the last two lines, which gives

$$\begin{aligned}\frac{\partial}{\partial\tau}F_s &= \sum_{s'}[R_{s'}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) - R_{s'}(\boldsymbol{x}^{\rm s}, 0)]\frac{\partial F_s}{\partial x_s}\\ &\quad + \sum_b[R_b(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) - R_b(\boldsymbol{x}^{\rm s}, 0)]\frac{\partial F_s}{\partial x_b}. \qquad \text{(H4)}\end{aligned}$$

This has the form of a Liouville equation as noticed in Ref. [18] and so its solution can be written as

$$F_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau) = F_s(\boldsymbol{\psi}^{\rm s}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau), \boldsymbol{\psi}^{\rm b}(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, \tau)), \qquad \text{(H5)}$$

where the components of the vector functions $\boldsymbol{\psi}^{\rm s}$ and $\boldsymbol{\psi}^{\rm b}$ evolve with τ according to

$$\frac{\partial}{\partial\tau}\psi_s = R_s(\boldsymbol{\psi}^{\rm s}, \boldsymbol{\psi}^{\rm b}) - R_s(\boldsymbol{\psi}^{\rm s}, 0), \qquad \text{(H6)}$$

$$\frac{\partial}{\partial\tau}\psi_b = R_b(\boldsymbol{\psi}^{\rm s}, \boldsymbol{\psi}^{\rm b}) - R_b(\boldsymbol{\psi}^{\rm s}, 0), \qquad \text{(H7)}$$

from the initial conditions

$$\psi_b(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, 0) = x_b, \qquad \psi_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}, 0) = x_s. \qquad \text{(H8)}$$

The function F_s at $\tau = 0$, which as before we write without a time argument, is given by the analog of Eq. (12):

$$F_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) = R_s(\boldsymbol{x}^{\rm s}, \boldsymbol{x}^{\rm b}) - R_s(\boldsymbol{x}^{\rm s}, 0). \qquad \text{(H9)}$$

The corresponding memory as defined in Eq. (19) is

$$M_s^{\rm G}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{s'} R_{s'}\frac{\partial}{\partial x_{s'}}F_s(\boldsymbol{\psi}^{\rm s}, \boldsymbol{\psi}^{\rm b}) + \sum_b R_b\frac{\partial}{\partial x_b}F_s(\boldsymbol{\psi}^{\rm s}, \boldsymbol{\psi}^{\rm b}), \qquad \text{(H10)}$$

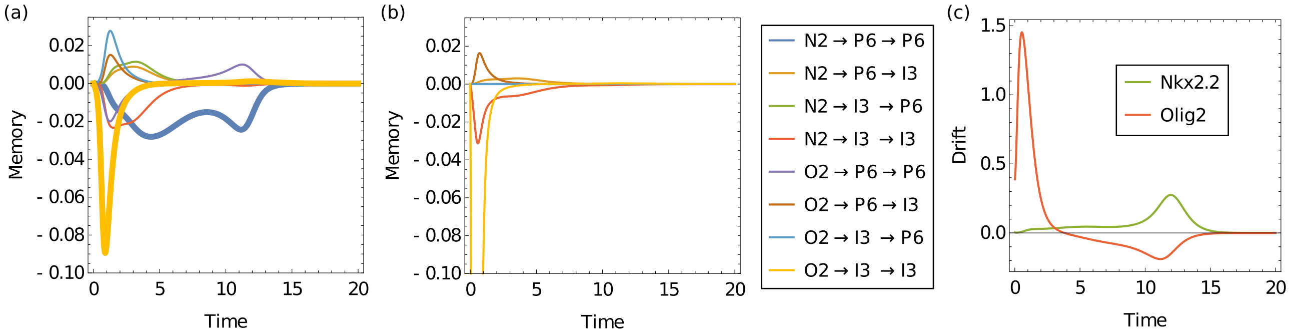

FIG. 13. Channel decomposition of the memory terms for the ZMs trajectory shown in Fig. 6. (a) and (b) show memory terms affecting Nkx2.2 and Olig2, respectively. Each color represents a memory channel as indicated. The memory originates from a particular subnetwork species "sending" memory through a specific bulk species; the effect then propagates within the bulk and returns via a specific bulk species (see legend). The two most salient memory functions are Nkx2.2 to Pax6 and then returning through Pax6 into Nkx2.2 (thick yellow line), and Olig2 to Irx3 and then returning through Pax6 into Nkx2.2 (thick blue line). (b) shows a large memory contribution acting on Olig2 via the channel through Irx3 (yellow line). However, in this case, the drift for Olig2 (c) is so large that the relative effect of this memory channel remains nonetheless small.

where all $R_{s'}$, R_b and the derivatives are evaluated at $(\boldsymbol{x}^{\rm s}, 0)$. Gouasmi *et al.* propose to find these derivatives numerically, but in fact a closed form expression can be obtained as follows. Applying the chain rule gives

$$M_s^{\rm G}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{s's''} R_{s'} \frac{\partial F_s}{\partial \psi_{s''}} \frac{\partial \psi_{s''}}{\partial x_{s'}} + \sum_{s'b} R_{s'} \frac{\partial F_s}{\partial \psi_b} \frac{\partial \psi_b}{\partial x_{s'}} + \sum_{s'b} R_b \frac{\partial F_s}{\partial \psi_{s'}} \frac{\partial \psi_{s'}}{\partial x_b} + \sum_{bb'} R_b \frac{\partial F_s}{\partial \psi_{b'}} \frac{\partial \psi_{b'}}{\partial x_b}. \tag{H11}$$

Now note that in the final evaluation we always use $\boldsymbol{x}^{\rm b} = 0$, which from the differential Eqs. (H6) and (H7) implies $\boldsymbol{\psi}^{\rm s} = \boldsymbol{x}^{\rm s}$, $\boldsymbol{\psi}^{\rm b} = 0$ for all τ. Hence, in particular ψ_b is independent of $\boldsymbol{x}^{\rm s}$ and so $\partial\psi_b/\partial x_{s'} = 0$. We also have $F_s(\boldsymbol{x}^{\rm s}, 0) = 0$ from Eq. (H9), which implies $\partial F_s/\partial\psi_{s'} = 0$. Only the last term from Eq. (H11) thus survives,

$$M^{\rm G}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{bb'} R_b \frac{\partial F_s}{\partial \psi_{b'}} \frac{\partial \psi_{b'}}{\partial x_b}, \tag{H12}$$

and it remains to find $\partial\psi_{b'}/\partial x_b$. By differentiating Eq. (H7) for $\partial\psi_{b'}/\partial\tau$ w.r.t. x_b, one finds

$$\frac{\partial}{\partial \tau} \frac{\partial \psi_{b'}}{\partial x_b} = \frac{\partial R_{b'}}{\partial \psi_{b''}} \frac{\partial \psi_{b''}}{\partial x_b}. \tag{H13}$$

On the r.h.s., a similar term from the variation of $\boldsymbol{\psi}^{\rm s}$ vanishes because it would be proportional to

$$\frac{\partial R_{b'}}{\partial \psi_s}(\boldsymbol{\psi}^{\rm s}, \boldsymbol{\psi}^{\rm b}) - \frac{\partial R_{b'}}{\partial \psi_s}(\boldsymbol{\psi}^{\rm s}, 0). \tag{H14}$$

This difference is zero in the final evaluation at $\boldsymbol{x}^{\rm b} = 0$ (which implies $\boldsymbol{\psi}^{\rm b} = 0$). For the same reason, the derivatives $\partial R_{b'}/\partial\psi_{b''} = \partial R_{b'}/\partial x_{b''}$ are evaluated at $(\boldsymbol{x}^{\rm s}, 0)$ and constant in time τ. Collecting these derivatives into a matrix $\boldsymbol{k}$ with elements $k_{b'b''}$ and using that $\partial\psi'_b/\partial x_b = \delta_{bb'}$ ($= 1$ for $b = b'$ and $= 0$ otherwise) at $\tau = 0$ gives then as the explicit solution of Eq. (H13),

$$\frac{\partial \psi_{b'}}{\partial x_b} = (e^{\boldsymbol{k}\tau})_{b'b}, \tag{H15}$$

and inserting into Eq. (H12) yields

$$M^{\rm G}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{bb'} \frac{\partial R_s}{\partial x_{b'}} (e^{\boldsymbol{k}\tau})_{b'b} R_b, \tag{H16}$$

where we have used that $\partial F_s/\partial\psi_{b'} = \partial F_s/\partial x_{b'} = \partial R_s/\partial x_{b'}$; this derivative and the factor R_b are evaluated at $(\boldsymbol{x}^{\rm s}, 0)$ in the approximation from Ref. [18] for the memory function.

We do not show here how the above memory approximation performs in our test systems because the nature of the approach can lead to fixed points disappearing after projection or new fixed points appearing. We observed both of these effects in numerical evaluations for the bistable switch from Ref. [30].

APPENDIX I: EXTENDING GOUASMI *ET AL.* APPROXIMATION WITH QSS PROJECTION

The Gouasmi *et al.* approximation [18] for the memory function rests on projecting to $\boldsymbol{x}^{\rm b} = 0$, but this is not generally an appropriate baseline for our case as it would correspond to setting all bulk concentrations to zero. However, we can adapt the approximation to the spirit of our work by changing coordinate system so that zero bulk coordinates correspond to the projection we consider throughout the paper, i.e., to QSS bulk concentrations. Explicitly, this variable transformation reads

$$\tilde{x}_s = x_s, \qquad \tilde{x}_b = x_b - x_b^*(\boldsymbol{x}^{\rm s}) \tag{I1}$$

because $\tilde{x}_b = 0$ is then equivalent to $x_b = x_b^*(\boldsymbol{x}^{\rm s})$. The time evolution of the new variables follows as

$$\frac{d}{dt}\tilde{x}_s = \tilde{R}_s(\tilde{\boldsymbol{x}}^{\rm s}, \tilde{\boldsymbol{x}}^{\rm b}) = R_s(\tilde{\boldsymbol{x}}^{\rm s}, \boldsymbol{x}^{\rm b*} + \tilde{\boldsymbol{x}}^{\rm b}), \tag{I2}$$

$$\frac{d}{dt}\tilde{x}_b = \tilde{R}_b(\tilde{\boldsymbol{x}}^{\rm s}, \tilde{\boldsymbol{x}}^{\rm b}), \tag{I3}$$

$$= R_b(\tilde{\boldsymbol{x}}^{\rm s}, \boldsymbol{x}^{\rm b*} + \tilde{\boldsymbol{x}}^{\rm b}) + \sum_s \left[\sum_{b'} (\boldsymbol{J}^{-1})_{bb'} \frac{\partial R_{b'}}{\partial x_s} \right] \times R_s(\tilde{\boldsymbol{x}}^{\rm s}, \boldsymbol{x}^{\rm b*} + \tilde{\boldsymbol{x}}^{\rm b}), \tag{I4}$$

where the factors enclosed in square brackets are the explicit expression for $-\partial x_b^*/\partial x_s$ and have to be evaluated at $\boldsymbol{x}^{\rm b*}$.

The Gouasmi memory function approximation, adapted for our QSS projection, is now given by Eq. (H16) applied to the new variables $\tilde{x}_s$, $\tilde{x}_b$ and corresponding drift functions $\tilde{R}_s$, $\tilde{R}_b$. The last factor is $\tilde{R}_b(\tilde{\boldsymbol{x}}^{\rm s}, 0)$, which can be read off from Eq. (I4). The first term in Eq. (I4) vanishes as $R_b = 0$ at QSS, while the remainder is seen to be precisely $c_b(\boldsymbol{x}^{\rm s})$ from Eq. (21). The matrix $\boldsymbol{k}$ in the new variables has elements

$$k_{b'b} = \frac{\partial \tilde{R}_{b'}}{\partial \tilde{x}_b} = \frac{\partial R_{b'}}{\partial x_b} + \sum_s \left[\sum_{b'} (\boldsymbol{J}^{-1})_{bb'} \frac{\partial R_{b'}}{\partial x_s} \right] \frac{\partial R_s}{\partial x_b} = l_{bb'}. \tag{I5}$$

Note that the terms in square brackets are already just dependent on $\boldsymbol{x}^{\rm s}$, so do not contribute to the derivative. The remaining factor in the memory function is, again in the new variables,

$$\frac{\partial \tilde{R}_s}{\partial \tilde{x}_{b'}} = \frac{\partial R_s}{\partial x_{b'}}, \tag{I6}$$

so, overall,

$$\tilde{M}_s^{\rm G}(\boldsymbol{x}^{\rm s}, \tau) = \sum_{bb'} \frac{\partial R_s}{\partial x_{b'}} (e^{\boldsymbol{l}(\boldsymbol{x}^{\rm s})\tau})_{bb'} c_b(\boldsymbol{x}^{\rm s}) = \sum_{b'} c_{b'}(\boldsymbol{x}^{\rm s}) \tilde{f}_{b's}(\boldsymbol{x}^{\rm s}, \tau) \tag{I7}$$

with

$$\tilde{f}_{b's}(\boldsymbol{x}^{\rm s}, \tau) = \left(e^{\boldsymbol{l}(\boldsymbol{x}^{\rm s})\tau}\right)_{b'b} f_{bs}^0(\boldsymbol{x}^{\rm s}), \tag{I8}$$

where we have used the definition of f_{bs}^0 from Eq. (14). Comparing now Eqs. (18) and (I8) shows that the memory approximation Eq. (I7), though derived here from rather different arguments, is quite similar to our expression Eq. (20): The only difference is that the propagation from $\boldsymbol{x}^{\rm s}$ to $\boldsymbol{\phi}_v(\boldsymbol{x}^{\rm s}, \tau)$ is absent in $\tilde{f}_{b's}$, which is the analog of our f_{bs}. We show that without the $\boldsymbol{\phi}_v$ propagation, the method can still perform accurately in some situations but breaks down in other settings (Fig. 10).

APPENDIX J: NEURAL TUBE MODEL

We detail the model taken from Ref. [24] used to model ventral neural tube patterning. The equations are as follows:

$$\frac{d}{dt}x_{\mathrm{P}} = \frac{\alpha_{\mathrm{P}} w_{\mathrm{P,p}}}{w_{\mathrm{P,p}} + (1 + k_{\mathrm{PO}} x_{\mathrm{O}})^2 (1 + k_{\mathrm{PN}} x_{\mathrm{N}})^2} - \beta_{\mathrm{P}} x_{\mathrm{P}}, \tag{J1}$$

$$\frac{d}{dt}x_{\mathrm{O}} = \frac{\alpha_{\mathrm{O}} w_{\mathrm{O,p}} (1 + k_{\mathrm{O,in}} x_{\mathrm{in}})}{w_{\mathrm{O,p}} (1 + k_{\mathrm{O,in}} x_{\mathrm{in}}) + (1 + k_{\mathrm{OI}} x_{\mathrm{I}})^2 (1 + k_{\mathrm{ON}} x_{\mathrm{N}})^2} - \beta_{\mathrm{O}} x_{\mathrm{O}}, \tag{J2}$$

$$\frac{d}{dt}x_{\mathrm{N}} = \frac{\alpha_{\mathrm{N}} w_{\mathrm{N,p}} (1 + k_{\mathrm{N,in}} x_{\mathrm{in}})}{w_{\mathrm{N,p}} (1 + k_{\mathrm{N,in}} x_{\mathrm{in}}) + (1 + k_{\mathrm{NP}} x_{\mathrm{P}})^2 (1 + k_{\mathrm{NO}} x_{\mathrm{O}})^2 (1 + k_{\mathrm{NI}} x_{\mathrm{I}})^2} - \beta_{\mathrm{N}} x_{\mathrm{N}}, \tag{J3}$$

$$\frac{d}{dt}x_{\mathrm{I}} = \frac{\alpha_{\mathrm{I}} w_{\mathrm{I,p}}}{w_{\mathrm{I,p}} + (1 + k_{\mathrm{IO}} x_{\mathrm{O}})^2 (1 + k_{\mathrm{IN}} x_{\mathrm{N}})^2} - \beta_{\mathrm{I}} x_{\mathrm{I}}. \tag{J4}$$

The parameters and their meaning in a biological sense are detailed as follows:

Name	Meaning	Value
α_{P}	Pax6 production rate	2
α_{O}	Olig2 production rate	2
α_{N}	Nkx2.2 production rate	2
α_{I}	Irx3 production rate	2
β_{P}	Pax6 degradation rate	2
β_{O}	Olig2 degradation rate	2
β_{N}	Nkx2.2 degradation rate	2
β_{I}	Irx3 degradation rate	2
k_{PO}	Olig2 binding to Pax6 DNA	1.9
k_{PN}	Nkx2.2 binding to Pax6 DNA	26.7
k_{ON}	Nkx2.2 binding to Olig2 DNA	60.6
k_{OI}	Irx3 binding to Olig2 DNA	28.4
k_{NP}	Pax6 binding to Nkx2.2 DNA	4.8
k_{NO}	Olig2 binding to Nkx2.2 DNA	27.1
k_{NI}	Irx3 binding to Nkx2.2 DNA	47.1
k_{IO}	Olig2 binding to Irx3 DNA	58.8
k_{IN}	Nkx2.2 binding to Irx3 DNA	76.2
$w_{\mathrm{P,p}}$	Polymerase binding to Pax6 DNA	3.84
$w_{\mathrm{O,p}}$	Polymerase binding to Olig2 DNA	2.01263
$w_{\mathrm{N,p}}$	Polymerase binding to Nkx2.2 DNA	0.572324
$w_{\mathrm{I,p}}$	Polymerase binding to Irx3 DNA	18.72
$k_{\mathrm{O,in}}$	Gli (Shh signal) binding to Olig2 DNA	180
$k_{\mathrm{N,in}}$	Gli (Shh signal) binding to Nkx2.2 DNA	373

The signal input concentration x_{in} is the gradient $e^{-p/0.15}$, which depends on the dorsal-ventral neural tube position p ranging from 0 to 1 as in Ref. [24].

[1] E. H. Davidson, Nature **468**, 911 (2010).

[2] J. Snider, M. Kotlyar, P. Saraon, Z. Yao, I. Jurisica, and I. Stagljar, Mol. Syst. Biol. **11**, 848 (2015).

[3] G. Rega and H. Troger, Nonlinear Dyn. **41**, 1 (2005).

[4] D. Schnoerr, G. Sanguinetti, and R. Grima, J. Phys. A: Math. Theor. **50**, 093001 (2017).

[5] L. Bronstein and H. Koeppl, Phys. Rev. E **97**, 062147 (2018).

[6] S. Nakajima, Prog. Theor. Phys. **20**, 948 (1958).

[7] R. Zwanzig, Phys. Rev. **124**, 983 (1961).

[8] H. Mori, Prog. Theor. Phys. **33**, 423 (1965).

[9] K. Kawasaki and J. D. Gunton, Phys. Rev. A **8**, 2048 (1973).

[10] R. Zwanzig, *Nonequilibrium Statistical Mechanics* (Oxford University Press, Oxford, UK, 2001).

[11] A. J. Chorin, O. H. Hald, and R. Kupferman, Proc. Natl. Acad. Sci. **97**, 2968 (2000).

[12] A. J. Chorin, O. H. Hald, and R. Kupferman, Physica D **166**, 239 (2002).

[13] E. Weinan, T. Li, and E. Vanden-Eijnden, Proc. Natl. Acad. Sci. USA **105**, 7907 (2008).

[14] A. J. Chorin and P. Stinis, Commun. Appl. Math. Comput. Sci. **1**, 1 (2006).

[15] P. Stinis, Physica D **213**, 197 (2006).

[16] C. L. Beck, S. Lall, T. Liang, and M. West, *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference Shanghai, China* (IEEE, Piscataway, NJ, 2009), pp. 3282–3287.

[17] P. Thomas, R. Grima, and A. V. Straube, Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys. **86**, 041110 (2012).

[18] A. Gouasmi, E. J. Parish, and K. Duraisamy, Proc. R. Soc. A **473**, 20170385 (2017).

[19] J. Xing and K. S. Kim, J. Chem. Phys. **134**, 044132 (2011).

[20] A. Mukhopadhyay and J. Xing, arXiv:1304.4908.

[21] K. J. Rubin, K. Lawler, P. Sollich, and T. Ng, J. Theor. Biol. **357**, 245 (2014).

[22] B. Bravi and P. Sollich, J. Stat. Mech.: Theory Exp. (2017) 063404.

[23] H. W. Kang, W. R. KhudaBukhsh, H. Koeppl, and G. A. Rempała, Bull. Math. Biol. **81**, 1303 (2019).

[24] M. Cohen, K. M. Page, R. Perez-Carrasco, C. P. Barnes, and J. Briscoe, Development **141**, 3868 (2014).

[25] K. Kawasaki, Ann. Phys. **61**, 1 (1970).

[26] W. Götze, *Liquids, Freezing and Glass Transition. Les Houches Summer Schools of Theoretical Physics Session LI* (North-Holland, Amsterdam, 1991), p. 287.

[27] E. Herrera-Delgado, R. Perez-Carrasco, J. Briscoe, and P. Sollich, PLoS Comput. Biol. **14**, e1006003 (2018).

[28] D. Angeli, J. E. Ferrell, and E. D. Sontag, Proc. Natl. Acad. Sci. **101**, 1822 (2004).

[29] K. M. Page and R. Perez-Carrasco, J. R. Soc., Interface **15**, 20180157 (2018).

[30] T. Wilhelm, BMC Syst. Biol. **3**, 90 (2009).

[31] D. Kondepudi and I. Prigogine, *Modern Thermodynamics: From Heat Engines to Dissipative Structures* (John Wiley and Sons, New Jersey, 1998).

[32] A. O. Caldeira and A. J. Leggett, Ann. Phys. **149**, 374 (1983).

[33] R. Zwanzig, J. Stat. Phys. **9**, 215 (1973).

[34] T. Enver, M. Pera, C. Peterson, and P. W. Andrews, Cell Stem Cell **4**, 387 (2009).

[35] M. B. Elowitz and S. Leibier, Nature **403**, 335 (2000).

[36] N. Strelkowa and M. Barahona, J. R. Soc., Interface **7**, 1071 (2010).

[37] A. Sagner and J. Briscoe, Development **146**, dev182154 (2019).

[38] K. Exelby, E. Herrera-Delgado, L. Garcia Perez, R. Perez-Carrasco, A. Sagner, V. Metzis, P. Sollich, and J. Briscoe, https://www.biorxiv.org/content/10.1101/721043v1 (2019).

[39] E. Dessaud, L. L. Yang, K. Hill, B. Cox, F. Ulloa, A. Ribeiro, A. Mynett, B. G. Novitch, and J. Briscoe, Nature **450**, 717 (2007).

[40] J. A. Sanders, F. Verhulst, and J. Murdock, *Averaging Methods in Nonlinear Dynamical Systems*, Vol. 59 (Springer, New York, 2007).

[41] F. John, *Partial Differential Equations*, Applied Mathematical Sciences, Vol. 1 (Springer US, New York, 1978).