DYNAMICAL EQUIVALENCE OF NETWORK ARCHITECTURE FOR COUPLED DYNAMICAL SYSTEMS I: ASYMMETRIC INPUTS

N AGARWAL AND M FIELD

Abstract. We give a simple necessary and sufficient condition for the dynamical equivalence of two coupled cell networks with different network architectures. The results are applicable to both continuous and discrete dynamical systems and are framed in terms of what we term input and output equivalence. We also give an algorithm that allows explicit construction of the cells in a system with a given network architecture in terms of the cells from an equivalent system with different network architecture. Details of proofs are provided for the case of cells with asymmetric inputs — details for the case of symmetric inputs are provided in a companion paper.

1. Introduction

Networks are used as models in a wide range of applications in biology, physics, chemistry, engineering and the social sciences (for many characteristic examples, we refer to the survey by Newman [12]). Of particular interest, especially in biology and engineering, are networks of interacting dynamical systems. Following the work of Kuramoto on networks of coupled phase oscillators with all-to-all coupling [11], there has been an intensive effort to understand the dynamical behavior of networks in terms of invariants of the network and to find conditions that imply the emergence of synchronization in complex networks. Typically the methods used for large complex networks are statistical and allow for the interaction of structurally identical units which have parameters (for example, coupling strength) and network connections distributed according to a statistical law. For a characteristic illustration of this approach, we refer to the article by Restrepo *et al.* [13] where conditions on the adjacency matrix of a network are

Date: February 16, 2010.
2010 *Mathematics Subject Classification.* 37C05, 37C10, 37C15.
Research of MF and NA supported in part by NSF Grants DMS-0600927 & DMS-0806321.

shown to lead to synchronization of a network. In a rather different direction, methods from symmetric dynamics have been used to understand symmetrically coupled networks of *identical* oscillators. One of the early works in this area is due to Ashwin & Swift [4] who study the dynamics of weakly coupled identical oscillators. More recently, Stewart, Golubitsky and coworkers [14, 7, 9] have formulated a general theory for networks of interacting (typically, identical) dynamical systems (for a overview, see [8]). Typically no symmetry is assumed for the network architecture though local symmetries may be present and these are described using groupoid formalism. While this type of model is unlikely to apply exactly to large biological networks, such as neuronal networks, it is possible that the assumption of identical dynamical system may be applicable to an 'averaged' network — that is, after the addition of noise and in the regime where there is synchronization. In any case, small (asymmetric) networks of identical dynamical systems typically display interesting and often quite nongeneric dynamics [3, 8] and there is the significant question concerning the extent to which the large networks that occur in say biology or engineering applications can be modelled in a hierarchical way as a network of small networks or *motifs* [10].

In this work we focus on the question of when two networks with apparently quite different topologies or architectures are dynamically equivalent (we give the precise definition shortly — we do not mean topological equivalence in the sense of conjugacy). We also consider and solve the problem of the explicit realization of equivalence for continuous dynamics and give a partial solution for discrete dynamics. These results are probably of greatest interest for relatively small networks. Indeed, although the invariants we describe are quite simple, they depend on the ordering of the cells and so checking of the equivalence of two networks potentially requires consideration of many different orderings. We emphasize that the methods we use involve precise descriptions of dynamics and are not statistical. The approach to networks we use in this work is synthetic and combinatorial in character. In particular, we adopt a simple and transparent 'flow-chart' formalism similar to that used in electrical and computer engineering. Indeed, ideas from analog computation motivate parts of our approach to networks (for more background, we refer to [6, 3]).

We view a network as a collection of interacting dynamical systems or 'cells'. The dynamics of a cell will be deterministic (specified by a vector field — continuous dynamics; or map — discrete dynamics). A *coupled cell system* will then be a specific set of individual but interacting cells. Each cell will have an *output* and a number of *inputs* coming from other cells in the system. An output might be the *state* of the cell (that is, the point in the phase space of the dynamical system which determines the evolution of the cell) or it could be a scalar or vector valued observable (for example, temperature and pressure or a membrane potential). Our setup is robust enough to handle both situations. Typically, if the network is small and cell dynamics are low dimensional, we assume the output is the state of the cell. For large networks or high dimensional cell dynamics, a vector valued observable is likely to be more appropriate. We emphasize that each cell has only one (type of) output. However, the output may be connected to many different inputs; these inputs do not have to be of the same type. (An analogy is having several different appliances powered by the same power socket.) Given a collection of cells there may be many different ways of connecting them into a network and the combinatorics of this process — viewed in a general way that allows groups of appropriately connected cells to define new types of cell — is one of the issues that is addressed in this paper.

A coupled cell system has a *network architecture* or *network structure* that can be represented by a directed graph with vertices corresponding to cells and each directed edge corresponding to a specific output–input connection. Different input types will correspond to different edge types in the graph. We use the term *coupled cell network* to refer to a network of coupled cell's with a specified network architecture. Thus vertices correspond to cells, edges to connections. If we want to emphasize a specific coupled cell network, we often use the term *coupled cell system* (see also section 2 for the notational conventions we use and note that we use the same symbol for both network architecture and coupled cell network).

Suppose that $\mathcal{M}$ and $\mathcal{N}$ are coupled cell networks which both have n cells. Assume that cells are modelled by vector fields (that is, continuous dynamics determined by local solution of ordinary differential equations) and that the phase space of each cell is a smooth manifold. We write $\mathcal{M} \prec \mathcal{N}$ ($\mathcal{M}$ is *dominated* by $\mathcal{N}$) if for every coupled cell system $\mathcal{F}$ with architecture $\mathcal{M}$, there exists a coupled cell system $\mathcal{F}^\star$ with architecture $\mathcal{N}$ such that $\mathcal{F}$ and $\mathcal{F}^\star$ have identical dynamics. This definition simply means that the dynamics of any system $\mathcal{F}$ with network architecture $\mathcal{M}$ can be realized by a system $\mathcal{F}^\star$ with network architecture $\mathcal{N}$. Implicit in the definition is the requirement that there is a correspondence between the cells of $\mathcal{F}$ and $\mathcal{F}^\star$ so that corresponding cells have identical phase space. We regard $\mathcal{M}$ and $\mathcal{N}$ as *equivalent* networks if $\mathcal{M} \prec \mathcal{N}$ and $\mathcal{N} \prec \mathcal{M}$. We may similarly define equivalence for networks modelled by discrete dynamics. Our main result (theorem 3.23) gives a simple necessary and sufficient condition for the equivalence of two networks in terms of an invariant that depends only on the network architecture (specifically, only on the adjacency matrices of the network). This result generalizes the linear equivalence results of Dias and Stewart [5] but the conceptual approach and methods are quite different. We emphasize that the algebraic condition, formulated in terms of adjacency matrices, is simple and easy to check — at least if we are given an ordering of the cells. However, realizing the equivalence using an output or input equivalence is by no means straightforward, especially when there are symmetric inputs (see [1]). The second question we address concerns the relation $\mathcal{M} \prec \mathcal{N}$. If the coupled cell system $\mathcal{F}$ has architecture $\mathcal{M}$, we present an explicit algorithm that allows us to construct a coupled cell system $\mathcal{F}^\star$ with architecture $\mathcal{N}$ such that $\mathcal{F}^\star$ has identical dynamics to $\mathcal{F}$. *The cells in the coupled cell system $\mathcal{F}^\star$ are constructed using the cells of $\mathcal{F}$ together with some passive cells that either scale or add and subtract outputs or inputs.* We also give a number of equivalence results that hold for discrete dynamics and for various classes of phase space (such as connected Abelian groups). In order to carry out this program we introduce the ideas of *input* and *output* equivalence. The idea of input equivalence is motivated by linear systems theory and involves taking linear combinations of inputs (necessarily, cell outputs must be either vector valued observables or state spaces must be linear). Output equivalence is formulated in terms of linear combinations of outputs or linear combinations of vector fields. Output equivalence may or may not apply to discrete systems defined on nonlinear spaces but it does apply to discretizations of ordinary differential equations.

In more detail, we start in section 2 with basic definitions and notational conventions. Our aim is to get the language and results as transparent as we can and, as far as possible, hide the (notational) complexities in the proofs of the results. In section 3, we introduce the concepts of input and output equivalence. We prove the main theorems on equivalence under the assumption that cells have asymmetric inputs (we present the more complex proofs for symmetric inputs in a companion paper [1]). Although equivalence always implies output equivalence for continuous systems, this is *not* always the case for input equivalence. In [1] we give necessary and sufficient conditions for equivalence to imply input equivalence. In section 4, we present examples that illustrate some outstanding issues for discrete networks. We also include a very simple, yet non-trivial, example illustrating the case where there are symmetric inputs.

We conclude by briefly describing the relation of our work to earlier results in this area. As we indicated above, a general theory of networks of coupled cells has been developed by Stewart, Golubitsky and coworkers. Their approach is relatively algebraic in character and strongly depends on groupoid formalism, graphs and the idea of a quotient network. Dias and Stewart [5] define equivalence in this setting and prove results on equivalence of networks when the phase space is a vector space. We provisionally refer to the definition of Dias and Stewart as 'functional equivalence'. It is easy to see that functional equivalence implies dynamical equivalence. The converse is also true since dynamical equivalence implies dynamical equivalence with phase space $\mathbb{R}$. Using the results of Dias and Stewart [5], dynamical equivalence with phase space $\mathbb{R}$ implies linear equivalence which implies functional equivalence. The methods of Dias and Stewart do not apply to systems for which the phase space is a general manifold nor do their methods give algorithms for realizing the equivalence. On account of their use of invariant theory and Schwarz' theorem on smooth invariants, they assume the phase space is linear and maps are smooth (that is C^∞). The methods we present in this work make no use of smooth invariant theory and indeed give a relatively elementary proof of the main results in [5].

2. Generalities on coupled cell networks

We distinguish between a *coupled cell network*, an abstract arrangement of cells and connections, and a *coupled cell system* which is a particular realization of a coupled cell network as a system of coupled dynamical equations. If we wish to emphasize the network graph rather than the dynamic structure, we refer to the *network architecture*. In this work we shall be particularly interested in networks with specific network architecture which satisfy additional constraints. Typically these constraints will relate to either the phase space or the type of input or output or the type of dynamics (for example, continuous or discrete).

We let $\mathbb{Z}$ denote the integers, $\mathbb{Z}^+$ denote the non-negative integers, $\mathbb{N}$ denote the strictly positive integers, and $\mathbb{Q}$ the rational numbers. If $k \in \mathbb{N}$, we use the abbreviated notation $\mathbf{k} = \{1, \cdots, k\}$. If $p \in \mathbb{N}$, then $\mathbf{k}^p$ denotes the set of all p-tuples $(k_1, \cdots, k_p)$, where $k_j \in \mathbf{k}$, for all $j \in \mathbf{p}$.

2.1. Structure of coupled cell networks: cells and connections.

For the moment we regard a cell as a 'black box' that admits various types of input (from other cells) and which has an output which is uniquely determined by the inputs and the initial state of the cell. The output may vary in discrete or continuous time. (Various interpretations are possible for the output; we refer to the introduction and later this section.) Two cells are regarded as being of the same *class* or *identical* if the same inputs and initial state always result in the same output. In this paper we largely restrict to networks of *identical* cells and leave the simple and straightforward extensions to more general networks containing different types of cell to the remarks (see also [6, 3]). For clarity, we always use the word *class* in the sense used above: 'two cells are of the same (or different) class'. We restrict the use of the word *type* to distinguish inputs of a cell.

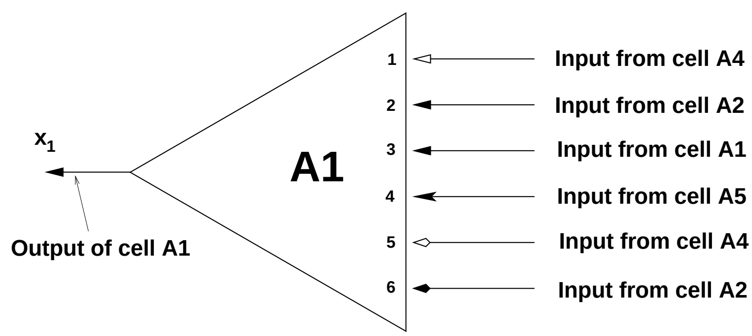
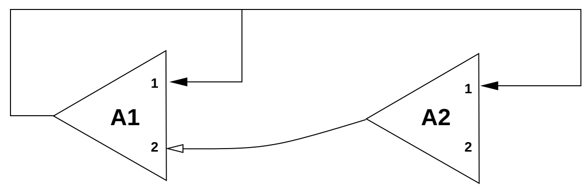


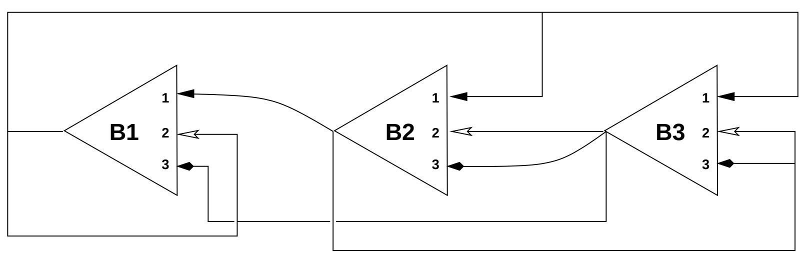
FIGURE 1. A cell with six inputs and one output. Inputs of the same type (for example the second and third input) can be permuted without affecting the output.

Figure 1 shows a cell, labelled **A1**, which accepts six inputs from cells **A1**, **A2**, **A4** and **A5**. We assume the cells **A2**, **A4** and **A5** are of the same class as **A1**. We denote the output of **A1** by x_1 and regard x_1 as specifying the *state* of the cell **A1** (later we shall vary this definition of output). Generally we do not regard inputs as interchangeable and we may distinguish different types of input by, for example, using different arrow heads. Referring to the figure, inputs 2 and 3 are of the *same type* whereas inputs $1, 4, 5$ and 6 are of different types and of different type from inputs 2 and 3: the cell has five distinct input types. We can interchange inputs 2 and 3 without changing the behaviour of the cell. We refer to these inputs as *symmetric*. If there are no pairs of symmetric inputs, we say that the cell has *asymmetric inputs*. In this work we only give proofs of results that assume asymmetric inputs (for the case of symmetric inputs see [1]).

We may think of cells as being coupled together using ‘patchcords’. Each patchcord goes from the output of a cell to the input of the same or another cell. We show two simple examples using identical cells in figure 2.



(i)



(ii)

FIGURE 2. Examples of coupled cells: (i) shows an incomplete network with an unfilled input cell **A2**, (ii) shows a three cell asymmetric input network; all inputs are filled.

When all the inputs of all the cells are filled, as they are in figure 2(ii), we refer to the set of cells and connections as a *network of coupled cells*.

We now give a formal definition of a coupled cell network based on the approach in Aguiar *et al.* [3].

Definition 2.1. A *coupled (identical) cell network* $\mathcal{N}$ consists of a finite number of identical cells such that

(a) The cells are patched together according to the input-output rules described above.
(b) There are no unfilled inputs.

Remarks 2.2. (1) We always assume cells have at least one input.
(2) There are no restrictions on the number of outputs we take from a cell.
(3) If a cell has multiple inputs of the *same* type (that is, symmetric inputs), it is immaterial which input of the symmetric set the patch-cord is plugged into. More precisely, if a cell **A** in the network has $k > 1$ inputs of the same type, then permutation of the k connections to these inputs is allowed and will not change the network structure. We emphasize that in this paper proofs are only given for the case of asymmetric inputs where the inputs to every cell in the network are of different type. When it comes to graphical representation of networks, we always represent input types to cells in the same order. If there are symmetric inputs, these are always grouped together (as in figure 1).
(4) A coupled cell network determines an associated directed graph (the *network architecture*) where the vertices of the graph are the cells and there is a directed edge from cell **X** to cell **Y** if and only if cell **Y** receives an input from cell **X**. Different input types will correspond to different edge types in the graph. If there are p different input types, then there will be p different edge types in the associated graph.

2.2. **Adjacency matrices of a network.** As we shall see, the key invariant of a coupled cell network is defined using the set of *adjacency* matrices. We recall the definition[1] appropriate to our context. Let **A** be a cell class. We suppose that **A** has r inputs and p input types. Let **A** have r_ℓ inputs of type ℓ, for $\ell \in \mathbf{p}$. Necessarily $r_1 + \cdots + r_p = r$. Of course, we assume $r_\ell \geq 1$. The cell **A** has asymmetric inputs iff $p = r$ and then $r_\ell = 1$, $\ell \in \mathbf{p}$. Suppose that $\mathcal{N}$ is a coupled cell network consisting of n cells $C_1, \cdots, C_n$ each of class **A**. We define $n \times n$ matrices $N_0, \cdots, N_p$. We take N_0 to be the identity matrix[2]. For $\ell \in \mathbf{p}0$, we let $N_\ell = [n^\ell_{ij}]$ be the matrix defined by $n^\ell_{ij} = k$ if there are exactly k inputs of type ℓ to C_j from the cell C_i. If there are no inputs of type ℓ from C_i, then $k = 0$. We refer to N_ℓ as the *adjacency matrix of type* ℓ for $\mathcal{N}$. Observe that the jth *column* of N_ℓ identifies the source cells for all the inputs of type ℓ to the cell C_j. If **A** has asymmetric inputs, then there are $r + 1$ adjacency matrices, $N_0, \cdots, N_r$ and each adjacency matrix will be a $0 - 1$ matrix with column sum equal to 1. If there are symmetric inputs, then there will be $p + 1 < r + 1$ adjacency matrices. The column sum of N_ℓ gives the

[1]Conventions vary. We choose the definition commonly used in graph theory; others take the transpose of the matrices we define.

[2]Strictly speaking, we only include N_0 if we allow internal variables — that is, the evolution of the state of the cell depends on its state, not just its initial state.

number of inputs of type ℓ to $\mathbf{A}$ and we refer to the column sum as the *valency* of N_ℓ. We denote the valency of N_ℓ by $\nu(\ell)$ and remark that $\nu(\ell) = \sum_{i=1}^n n_{ij}^\ell = r_\ell$ (independent of $j \in \mathbf{n}$). Let $\mathbb{A}(\mathcal{N})$ denote the (ordered) set $\{N_0, \cdots, N_p\}$ of adjacency matrices of $\mathcal{N}$.

Remarks 2.3. (1) If we allow multiple cell classes, then we define an adjacency matrix for each input type of every cell class.
(2) If we deny self-loops in the network structure, then the diagonal entries of the adjacency matrices $N_1, \cdots, N_p$ will all be zero.

The adjacency matrices N_1, N_2, N_3 for the network of figure 2(ii) are shown below.

$$N_1 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ N_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},\ N_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Assume for the moment that $\mathcal{M}$ is a coupled cell network consisting of n identical cells with r *asymmetric inputs* (the number r of inputs equals the number of input types p). Label the cells as $\mathbf{C}_1, \cdots, \mathbf{C}_n$. For each cell $\mathbf{C}_j$, we let $\mathfrak{m}^j = (\mathfrak{m}_1^j, \cdots, \mathfrak{m}_r^j) \in \mathbf{n}^p$ denote the r-tuple defined by requiring that there is an output from $\mathbf{C}_{\mathfrak{m}_i^j}$ to input i of $\mathbf{C}_j$. That is, $\mathfrak{m}_i^j$ identifies the source cell for the input of type i to C_j. If we denote the adjacency matrices of $\mathcal{M}$ by $M_0, \cdots, M_r$, then $\mathfrak{m}_i^j$ is the row index of the unique non-zero entry in column j of M_i. The $r \times n$ matrix $\mathfrak{m} = [\mathfrak{m}^1, \cdots, \mathfrak{m}^n]$ specifies the complete set of all connections to the cells in $\mathcal{M}$. We refer to $\mathfrak{m}$ as the *connection matrix* of the network $\mathcal{M}$. We emphasize that in order to define uniquely the connection matrix (and adjacency matrices) of a network, we need to order the cells and the input types. For future reference, note that if V is any vector space, then

$$\sum_{i=1}^n m_{ij}^0 x_i = x_j, \qquad \sum_{i=1}^n m_{ij}^\ell x_i = x_{\mathfrak{m}_\ell^j},\ \ell \in \mathbf{r},\ j \in \mathbf{n}, \tag{2.1}$$

where $x_1, \cdots, x_n \in V$ and $M_\ell = [m_{ij}^\ell]$, $\ell \in \{0, \cdots, r\}$, are the adjacency matrices. The result is obvious if $\ell = 0$, so suppose $\ell \in \mathbf{r}$. Then $m_{\alpha j}^\ell \neq 0$ iff C_j has an input of type ℓ from C_α. If this is so, then $m_{\alpha j}^\ell = 1$ and $m_{ij}^\ell = 0$, $i \neq \alpha$ (since we assume asymmetric inputs). Hence $\sum_{i \in \mathbf{n}} m_{ij}^\ell x_i = m_{\alpha j}^\ell x_\alpha = x_\alpha = x_{\mathfrak{m}_\ell^j}$, by definition of $\mathfrak{m}_\ell^j$.

2.3. **Discrete and continuous coupled cell systems.** We now define two basic classes of coupled cell networks with specified network architecture. First some notational conventions. If $\mathcal{N}$ denotes a network architecture, then by $\mathcal{F} \in \mathcal{N}$ we mean that $\mathcal{F}$ is a coupled cell system with connection and input type structure given by $\mathcal{N}$. If $\mathcal{N}$ is a coupled cell network (viewed as the collection of all coupled cell systems with network architecture $\mathcal{N}$), then the number of cells $n = n(\mathcal{N})$, the number of input types $p = p(\mathcal{N})$, and the total number of inputs $r = r(\mathcal{N})$, are the same for all systems $\mathcal{F} \in \mathcal{N}$.

Continuous dynamics modelled by ordinary differential equations. For continuous dynamics, we assume that cell outputs (and therefore inputs) depend continuously on time. The standard model for this situation is where each cell is modelled by an (autonomous) ordinary differential equation. In this work, we always assume that the evolution in time of cells depends on their internal state (not just their initial state). The underlying phase space for a cell will be a smooth manifold M, often $\mathbb{R}^N$, $N \geq 1$, or $\mathbb{T}$ (the unit circle). We assume the associated vector field has sufficient regularity to guarantee unique solutions.

If the phase space for a cell $\mathbf{C}$ is M, then the output $x(t)$ of $\mathbf{C}$ at time t defines a smooth curve in M and $x(0)$ will be the initial state of the cell (at time $t = 0$). We identify $x(t)$ with the internal state of the cell. If the cell has no inputs, then the ordinary differential equation model for the cell is $x' = f(x)$, where f is a vector field on M.

Suppose that we are given a coupled cell system $\mathcal{F} \in \mathcal{M}$ with identical cells $\{\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_n\}$. Assume that cells are modelled by ordinary differential equations and that each cell has r (asymmetric) inputs. For $j \in \mathbf{n}$, the dynamics of $\mathbf{C}_j$ will be given by a differential equation

$$x'_j = f(x_j; x_{\mathfrak{m}^j_1}, x_{\mathfrak{m}^j_2}, \cdots, x_{\mathfrak{m}^j_r}),$$

where $\mathfrak{m} = [\mathfrak{m}^1, \cdots, \mathfrak{m}^n]$ is the connection matrix of the network (see the previous section). Observe that we always write the internal variable x_j as the first variable of the vector field f. Since cells are assumed identical, the vector field f is independent of $j \in \mathbf{n}$. We often say that the dynamics of the system $\mathcal{F}$ is *modelled by* f and refer to f as the *model* for $\mathcal{F}$. If we need to emphasize the dependence of the model f on the system $\mathcal{F}$, we write $f_{\mathcal{F}}$ rather than f. All of this terminology applies equally well to discrete systems (see below).

Remarks 2.4. (1) If there is no dependence on the internal variable, we omit the initial x_j and write $x'_j = f(x_{\mathfrak{m}^j_1}, x_{\mathfrak{m}^j_2}, \cdots, x_{\mathfrak{m}^j_r})$.

(2) We do not require that $\mathfrak{m}^j_1, \cdots, \mathfrak{m}^j_r$ are distinct integers; indeed they may all be equal. If there are symmetric inputs we group these together and designate the group by an overline. For example, if the vector field f is symmetric in the first k-inputs and asymmetric in the remaining inputs we write $f(x_j; \overline{x_{\mathfrak{m}^j_1}, \cdots, x_{\mathfrak{m}^j_k}}, x_{\mathfrak{m}^j_{k+1}}, \cdots, x_{\mathfrak{m}^j_r})$.

Example 2.5. A coupled cell system with the network architecture of figure 2(ii) is realized by the differential equations

$$\begin{aligned} x_1' &= f(x_1; x_2, x_1, x_3), \\ x_2' &= f(x_2; x_1, x_3, x_3), \\ x_3' &= f(x_3; x_1, x_2, x_2). \end{aligned}$$

where $f : M \times M^3 \rightarrow TM$ is a (smooth) family of vector fields on M, depending on parameters in M^3. That is, for each $(x, (y, z, u)) \in M \times M^3$, $f(x; y, z, u) \in T_x M$.

Discrete dynamics. We continue with the notation of the previous section. We define a discrete time coupled cell system by considering a system of coupled maps updated at regular time intervals.

Exactly as in the continuous time case, we model a cell $\mathbf{C}_j$ at time N using a phase space variable $x_j(N)$ and then update all cells simultaneously by

$$x_j(N+1) = f(x_j(N); x_{\mathfrak{m}_1^j}(N), x_{\mathfrak{m}_2^j}(N), \cdots, x_{\mathfrak{m}_r^j}(N)),\ j \in \mathbf{n}, \tag{2.2}$$

where f is a continuous or smooth function depending on the internal state $x(N)$ together with the r inputs to the cell.

Remarks 2.6. (1) The assumption of regular updates implies the existence of a synchronizing mechanism such as a clock. It is also possible (and very useful) to define *asynchronous* systems but we shall not develop that aspect of the theory in this article.
(2) The diagrammatic conventions we follow are reminiscent of the transfer function diagrams used in linear systems theory. However, we are working with nonlinear systems and not taking Laplace transforms. For discrete dynamics, we start at $t = 0$ with each cell initialized and then we update according to (2.2) at specific time increments $\delta > 0$. The case of continuous dynamics can be viewed as the limiting case as $\delta \rightarrow 0$ — indeed, ordinary differential equations are typically solved on a digital computer by replacing the continuous model by a discrete model with an appropriately small value of $\delta > 0$.
(3) In many cases it can be useful to combine both discrete and continuous dynamics in a coupled cell system. This is particularly so for models of fast-slow systems (prevalent for example in neural systems). We may also consider systems with thresholds controlling switching in cells where the dynamics are, for the most part, governed by (smooth) differential equations. We refer to this type of system as a *hybrid* coupled cell system. Again we shall not develop this theory here but see Aguiar et al [3] for definitions and some simple examples.

2.4. **Passive cells.** As we shall shortly see it is sometimes useful to incorporate 'passive' cells into a network with continuous or discrete dynamics. We give two examples that we shall use later.

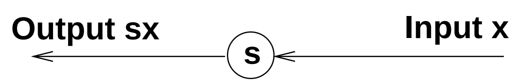


FIGURE 3. Scaling cell

Example 2.7 (Scale)**.** A *scaling* cell has one input. The input can either be a vector field or vector in $\mathbb{R}^N$. If the input is x then the output is sx, where $s \in \mathbb{R}$ is fixed. We use the notation shown in figure 3 for a scaling cell.

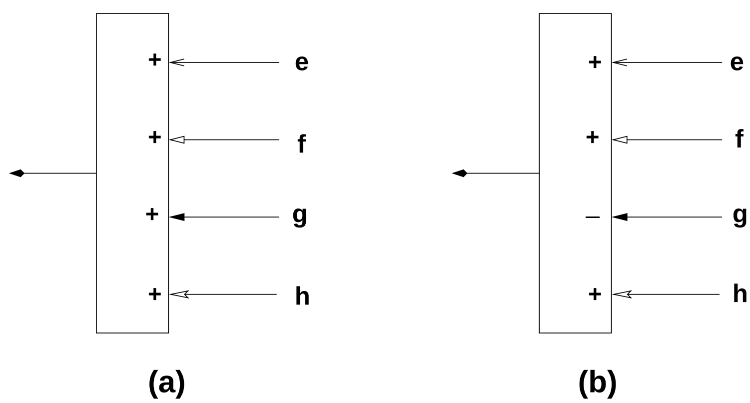


FIGURE 4. Add and Add-Subtract cells

Example 2.8 (Addition/Subtraction)**.** An *Add* cell has at least two inputs which must be either vector fields, points in $\mathbb{R}^N$ or, more generally, points in an Abelian Lie group. In figure 4(a) we show an Add cell with four inputs. If the inputs are $\mathbf{e}$, $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ then the output is $\mathbf{e}+\mathbf{f}+\mathbf{g}+\mathbf{h}$. We remark that an Add cell always has symmetric inputs though the symmetry of the inputs will not have implications for our intended applications. We can vary the picture a little by combining one or more inputs to an Add cell with a scale cell with $s = -1$. In this way we can do arbitrary addition and subtraction. We show in figure 4(b) an Add-Subtract cell with three additive inputs and one subtractive. The output of this cell is $\mathbf{e}+\mathbf{f}-\mathbf{g}+\mathbf{h}$. In the sequel, we use the notation shown in figure 4 for Add and Add-Subtract cells.

Example 2.9. We may use Add, Add-Subtract and scaling cells to build new cells from old. In figure 5 we show two characteristic examples. We start with a cell **C** which we suppose has two inputs. Assume

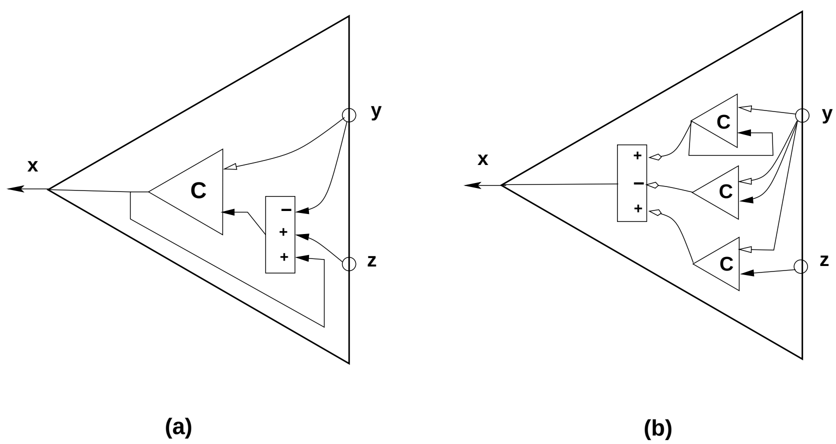


FIGURE 5. Building new cells using Add, Add-Subtract cells

a differential equation model with *linear* phase space $\mathbb{R}^N$. In figure 5(a) we show a new two-input cell constructed using **C** and an Add-Subtract cell. If dynamics on **C** is defined by $f : \mathbb{R}^N \times (\mathbb{R}^N)^2 \to \mathbb{R}^N$, then dynamics on the cell defined in figure 5(a) is given by $F : \mathbb{R}^N \times (\mathbb{R}^N)^2 \to \mathbb{R}^N$ where

$$F(x; y, z) = f(x; y, x - y + z).$$

This model would also work if the phase space for **C** was the N-torus $\mathbb{T}^N$. The same model also works for discrete cells provided that the phase space is either $\mathbb{R}^N$ or $\mathbb{T}^N$. However, the model does not work if the phase space for **C** is a general manifold — for example, the two-sphere S^2.

In figure 5(b) we combine outputs rather than inputs. The corresponding vector field for the cell shown in figure 5(b) is given by

$$F(x; y, z) = f(x; y, x) - f(x; y, y) + f(x; y, z).$$

This is valid for continuous dynamics on an arbitrary smooth manifold.

Remarks 2.10. (1) For discrete dynamics on a manifold, we can linearly combine outputs provided that $f(x_0; x_1, \ldots, x_r)$ is sufficiently close to x_0 for all $x_0, \ldots, x_r \in M$. For this it suffices to fix a Riemannian metric on M and then do addition and scalar multiplication in the tangent space $T_{x_0}M$ using the exponential map of the metric (that is,

$\exp_{x_0} : T_{x_0}M \to M$ is a local diffeomorphism on a neighborhood of the origin of $T_{x_0}M$). Note that this approach does not work for inputs as there is no reason to assume that linear combinations of $x_0, \ldots, x_r$ need be close to x_0. Once we have fixed a Riemannian metric on M, we can always linearly combine outputs for the discretization of the differential equation (at least if the time step is sufficiently small).
(2) For continuous dynamics, we interpret figure 5(b) in the following way: we assume the outputs of $\mathbf{C}$ are un-integrated — that is vector fields. We then linearly combine, scale and integrate to get the output. If we go to the discretization, then the previous remark holds (cf. remarks 2.6(2)).

2.5. **Some special classes of coupled cell networks.** Throughout this section, $\mathcal{M}$ will denote a fixed network architecture. We have already indicated that we also regard $\mathcal{M}$ as the set of all coupled cell systems with network architecture $\mathcal{M}$. Henceforth we assume that any coupled cell system $\mathcal{S} \in \mathcal{M}$ always has a phase space which is a smooth connected differential manifold.

It is appropriate to single out some special classes of coupled cell networks. The restrictions we impose are on the phase space and connection structure rather than on the type of the system (continuous, discrete, hybrid, etc).

(1) $\mathcal{M}(\mathbb{L})$ denotes the set of systems $\mathcal{S} \in \mathcal{M}$ which have linear phase space.
(2) $\mathcal{M}(\mathbb{T})$ denotes the set of systems for which the phase space is a compact connected Abelian Lie group — that is, an N-torus for some $N \geq 1$.
(3) $\mathcal{M}(\mathbb{G})$ denotes the set of systems for which the phase space is a Lie group.

We extend this notation to consider networks with specific phase spaces. For example, let $\mathcal{M}(\mathbb{R})$ denotes the set of systems $\mathcal{S} \in \mathcal{M}(\mathbb{L})$ which have phase space $\mathbb{R}$ and $\mathcal{M}(\mathrm{SO}(3))$ denote the class of systems $\mathcal{S} \in \mathcal{M}(\mathbb{G})$ which have phase space SO(3). We remark that $\mathcal{M} \supset \mathcal{M}(\mathbb{G}) \supset \mathcal{M}(\mathbb{L}), \mathcal{M}(\mathbb{T})$ and $\mathcal{M}(\mathbb{L}) \supset \mathcal{M}(\mathbb{R})$.

Scalar signalling networks. As we have remarked previously, from the point of view of applications it is unrealistic to assume that in a large network each cell has to have access to complete knowledge of the state of cells from which it receives outputs. (Of course, complete information may be important in small networks of cells with low dimensional phase spaces.) Suppose then that we have an identical cell system comprised of cells of class $\mathbf{C}$. Denote the phase space of $\mathbf{C}$ by M. Let

$\xi : M \rightarrow \mathbb{F}$ be a smooth function, where $\mathbb{F}$ denotes either the real or complex numbers[3]. If $x(t)$ is a trajectory on M, then $\xi(x(t))$ will be a curve in $\mathbb{F}$. While we could regard ξ as an observable, in our context we prefer to think of ξ a signal. For example, in neuronal dynamics, $\xi(x(t))$ might typically be zero or small except when the neuron spikes.

Definition 2.11. An identical coupled cell system $\mathcal{S} \in \mathcal{M}$ is a *scalar signalling system* if there exists a signal $\xi : M \rightarrow \mathbb{F}$ defined on the phase space of each cell such that the inputs to each cell depend only on the signals from the corresponding output cells. Let $\mathcal{M}(\mathbb{S})$ denote the class of scalar signalling systems $\mathcal{S} \in \mathcal{M}$.

Example 2.12. Let $\mathcal{S}$ be a scalar signalling system with the network of figure 2(ii). Let M be the phase space of a cell and $\xi : M \rightarrow \mathbb{F}$ be a signal. Set $\xi \circ x = \hat{x}$. With these notational conventions and a continuous differential equation model, the differential equations for the system will be

$$\begin{aligned} x_1' &= f(x_1; \hat{x}_2, \hat{x}_1, \hat{x}_3), \\ x_2' &= f(x_2; \hat{x}_1, \hat{x}_3, \hat{x}_3), \\ x_3' &= f(x_3; \hat{x}_1, \hat{x}_2, \hat{x}_2). \end{aligned}$$

We remark that the internal variables are *not* changed.

Remarks 2.13. (1) A key feature of scalar signalling systems is that we can linearly combine inputs even though the phase space may be nonlinear. In particular, the configuration shown in figure 5(a) is valid for both continuous *and* discrete scalar signalling systems irrespective of the phase space.
(2) The generalization of the definition of scalar signalling systems to networks with multiple cell classes is completely straightforward. This extension is significant as the main application we have in mind for scalar signalling systems is the coupling together of small networks (not scalar signalling) into large networks with scalar signalling between the small networks. In this context, it may well be appropriate to assume that there are no self loops round the small networks (see remark 2.3(2)) even though the cells within the small networks may have self loops. As a simple illustration, an audio amplifier internally may have several negative feedback loops but feeding an output of the complete audio system back into the audio source (say, a microphone) is generally not advisable. Similar observations apply in control theory.
(3) Finally, we will abuse notation and refer, for example, to a coupled cell network $\mathcal{M}(\mathbb{L})$. By this we mean that the network architecture

[3] More generally, $\mathbb{F}$ could be a finite dimensional vector space

is $\mathcal{M}$ but we will always restrict to systems with linear phase space. Similar remarks hold for the other network classes we have defined. The reason we do this is that we shall be defining various relations and orders on networks and some of these definitions only apply if we assume extra structure on the connections or the phase space.

3. Equivalence of networks

In this section we develop various notions of equivalence for coupled cell networks. We follow the approach of Aguiar *et al.* [3] and concentrate on dynamics rather than adopt the more abstract viewpoint of Dias and Stewart [5] which is based on groupoid formalism and restricted to systems with linear phase space (see below).

Suppose then that $\mathcal{F}$ and $\mathcal{G}$ are coupled cell systems which both have n cells (which need not be identical). Label the cells of $\mathcal{F}$ and $\mathcal{G}$ as $\{\mathbf{C}_1, \ldots, \mathbf{C}_n\}$ and $\{\mathbf{D}_1, \ldots, \mathbf{D}_n\}$ respectively. Assume that both systems have the same type of dynamics: either both discrete or both continuous. For ease of exposition we henceforth assume continuous dynamics modelled by ordinary differential equations but emphasize that most of what we say applies equally well to discrete dynamics. We indicate in the remarks when it does not.

The coupled cell systems $\mathcal{F}$, $\mathcal{G}$ have *identical dynamics* if

(1) The cells $\mathbf{C}_i$, $\mathbf{D}_i$ have the same phase space, $i \in \mathbf{n}$.
(2) The time evolution of both systems is identical.

Remarks 3.1. (1) If $\mathcal{F}$ and $\mathcal{G}$ have identical dynamics it does not follow that $\mathcal{F}$ and $\mathcal{G}$ have the same network architecture or that corresponding cells have the same number of inputs.
(2) Note that the definition of identical dynamics requires no restrictions on the type of phase space or connections.

We define a partial ordering on coupled cell networks, and an associated equivalence relation.

Definition 3.2 ([3])**.** Let $\mathcal{N}, \mathcal{M}$ be coupled cell networks both with n cells.

(a) The network $\mathcal{N}$ is *dominated* by $\mathcal{M}$, denoted $\mathcal{N} \prec \mathcal{M}$, if given an ordering of the cells in $\mathcal{N}$, we can choose an ordering of the cells of $\mathcal{M}$ such that given any system $\mathcal{F} \in \mathcal{N}$, there exists a system $\mathcal{F}^\star \in \mathcal{M}$ such that $\mathcal{F}$ and $\mathcal{F}^\star$ have identical dynamics.
(b) We say $\mathcal{N}$ and $\mathcal{M}$ are *equivalent*, denoted $\mathcal{N} \sim \mathcal{M}$, if we can order the cells in $\mathcal{M}$ and $\mathcal{N}$ so that $\mathcal{N} \prec \mathcal{M}$ and $\mathcal{M} \prec \mathcal{N}$.

Remarks 3.3. (1) In definition 3.2(b) it is not necessary to assume that the orderings of cells for the which $\mathcal{N} \prec \mathcal{M}$ and $\mathcal{M} \prec \mathcal{N}$ are the

same. Indeed, if they are not it is easy to see that we obtain a non-trivial permutation of the ordering of the first ordering of $\mathcal{M}$ relative to which $\mathcal{M} \sim \mathcal{M}$. Since we are assuming finitely many cells in $\mathcal{M}$, the order of the permutation is finite and from this we easily deduce that we can choose an ordering of the cells of $\mathcal{M}$ and $\mathcal{N}$ for which we have $\mathcal{N} \prec \mathcal{M}$ and $\mathcal{M} \prec \mathcal{N}$.
(2) If we restrict attention to systems with linear phase space, we can also define linear equivalence [5]. We write $\mathcal{M}(\mathbb{L}) \prec_L \mathcal{N}(\mathbb{L})$ if given $\mathcal{F} \in \mathcal{M}(\mathbb{L})$ modelled by a linear differential equation, there exists $\mathcal{F}^\star \in \mathcal{N}(\mathbb{L})$ modelled by a linear differential equation, such that $\mathcal{F}$ and $\mathcal{F}^\star$ have identical dynamics. We say the networks $\mathcal{N}$ and $\mathcal{M}$ are *linearly equivalent*, denoted by $\mathcal{M}(\mathbb{L}) \sim_L \mathcal{N}(\mathbb{L})$, if $\mathcal{M}(\mathbb{L}) \prec_L \mathcal{N}(\mathbb{L})$ and $\mathcal{N}(\mathbb{L}) \prec_L \mathcal{M}(\mathbb{L})$. (We remind the reader of the abuse of notation indicated in remarks 2.13(3).)

For the remainder of the section we assume identical cell networks.

For $n, m \in \mathbb{N}$, let $M(n,m;\mathbb{K})$ denote the space of $n \times m$-matrices with entries in $\mathbb{K}$ where $\mathbb{K}$ will be either $\mathbb{Q}$, $\mathbb{Z}$ or $\mathbb{Z}^+$. In case $m = n$, set $M(n,m;\mathbb{K}) = M(n;\mathbb{K})$.

Let $\mathcal{M}$ be an r-input n cell network with adjacency matrices $M_0 = I, M_1, \dots, M_p$. Let $\mathbf{A}(\mathcal{M})$ denote the vector subspace of $M(n;\mathbb{Q})$ spanned by $M_0, \dots, M_p$ and $\mathbf{A}(\mathcal{M};\mathbb{Z}^+)$ denote the set of all non-negative integer combinations of $M_0, M_1, \cdots, M_p$. We emphasize that to define the adjacency matrices we fix an ordering of the cells. In particular, the space $\mathbf{A}(\mathcal{M})$ will depend on the choice of ordering (but not on the ordering of input types).

We recall the main result of [5] adapted to our context and notational conventions.

Theorem 3.4 ([5])**.** *Let $\mathcal{M}, \mathcal{N}$ be coupled cell networks both with n cells. The following conditions are equivalent:*

1. $\mathcal{M}(\mathbb{L}) \prec \mathcal{N}(\mathbb{L})$. *(Phase space linear.)*
2. $\mathbf{A}(\mathcal{M}) \subset \mathbf{A}(\mathcal{N})$.
3. $\mathcal{M}(\mathbb{L}) \prec_L \mathcal{N}(\mathbb{L})$.
4. $\mathcal{M}(\mathbb{R}) \prec_L \mathcal{N}(\mathbb{R})$. *(Phase space $\mathbb{R}$).*

In particular, $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ *iff* $\mathcal{M}(\mathbb{L}) \sim \mathcal{N}(\mathbb{L})$.

Remarks 3.5. (1) In all statements it is assumed that there is a given ordering of cells in $\mathcal{N}$ and that we can choose an ordering of cells in $\mathcal{M}$ for which the corresponding statement holds.
(2) Theorem 3.4 does not apply to systems with non-linear phase space. Indeed, (3,4) have no meaning when the phase space is non-linear and the methods used in [5] do not apply to prove the equivalence of (1)

and (2) for the case of non-linear phase spaces.
(3) Theorem 3.4 applies when the network is governed by discrete dynamics and phase spaces are linear.

As a corollary of theorem 3.4 we have

Theorem 3.6. *Let $\mathcal{M}, \mathcal{N}$ be coupled cell networks both with n cells. A necessary condition for $\mathcal{M} \sim \mathcal{N}$ is $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$.*

Proof. If $\mathcal{M} \sim \mathcal{N}$ then $\mathcal{M}(\mathbb{L}) \sim \mathcal{N}(\mathbb{L})$ and so the result follows from theorem 3.4. □

Remark 3.7. It is easy to verify theorem 3.6 directly. Specifically, the difficult part of the proof of theorem 3.4 involves the verification that $\mathcal{M}(\mathbb{R}) \prec_L \mathcal{N}(\mathbb{R})$ implies $\mathcal{M}(\mathbb{L}) \prec \mathcal{N}(\mathbb{L})$. This implication is not needed for the proof of theorem 3.6.

3.1. **Input equivalence.** Input equivalence is applicable to coupled cell systems with linear phase space and to scalar signalling networks. The basic idea is that a network $\mathcal{N}$ 'input dominates' a network $\mathcal{M}$ if given a system $\mathcal{F} \in \mathcal{M}(\mathbb{L})$, we can find a system $\mathcal{G} \in \mathcal{N}(\mathbb{L})$ which has identical dynamics to $\mathcal{F}$ and is such that each cell in $\mathcal{G}$ is built from one cell in $\mathcal{F}$ together with a number of Add-Subtract and scaling cells acting on the inputs (see figure 5(a)). We start with a simple example to illustrate the ideas.

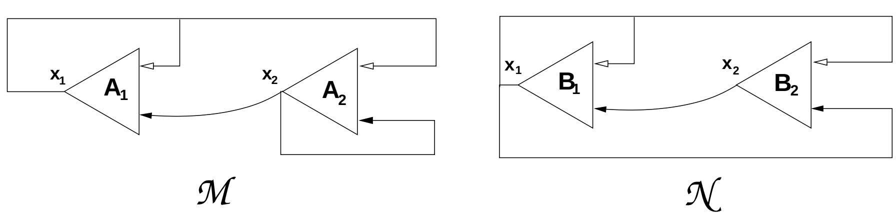


FIGURE 6

Example 3.8 ([3])**.** Referring to figure 6, suppose that $\mathcal{F} \in \mathcal{M}(\mathbb{L})$ is modelled by

$$\begin{aligned} x_1' &= f(x_1; x_1, x_2), \\ x_2' &= f(x_2; x_1, x_2), \end{aligned}$$

where $f = f_{\mathcal{F}} : V \times V^2 \to V$ is a C^1 function on the vector space V. If we define the C^1 model $g = g_{\mathcal{G}}$ for $\mathcal{G} \in \mathcal{N}(\mathbb{L})$ by

$$g(x_0; x_1, x_2) = f(x_0; x_1, x_0 - x_1 + x_2),$$

then

$$\begin{aligned} x_1' &= g(x_1; x_1, x_2) = f(x_1; x_1, x_2), \\ x_2' &= g(x_2; x_1, x_1) = f(x_2; x_1, x_2). \end{aligned}$$

Hence we can realize the dynamics of the first system $\mathcal{F}$ using the second system $\mathcal{G}$ which has a different architecture. Indeed, we can build the second network using Add-Subtract cells together with the cell used in $\mathcal{F}$. See figure 7 (and also figure 5(a)). Observe that the

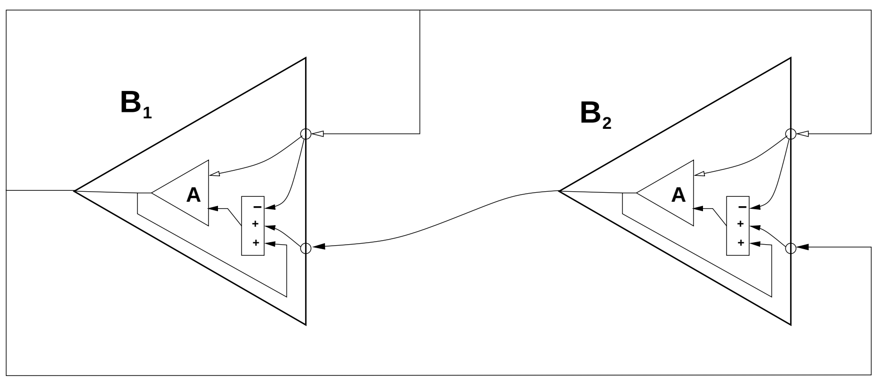


FIGURE 7. The system $\mathcal{G}$ built using the cell from $\mathcal{F}$ and an Add-Subtract cell.

system shown in figure 7 can be transformed back into the system $\mathcal{F}$ by removing the outer triangles defining cells of class **B** and then cancelling inputs using linearity. We say that the network $\mathcal{M}$ is *input dominated* by $\mathcal{N}$. Finally, the arguments above apply to scalar signaling networks (definition 2.11) — regard $f : V \times \mathbb{R}^2 \to V$ and define $g(x_0; \hat{x}_1, \hat{x}_2) = f(x_0; \hat{x}_1, \hat{x}_0 - \hat{x}_1 + \hat{x}_2)$.

We now extend the previous example and define the concepts of input domination and equivalence for general networks with asymmetric inputs. Conceptually, the idea is quite simple: one network is input dominated by another if the dynamics of any system in the first network can be realized by a system in the second network whose cells are constructed from those in the first network by linearly combining inputs.

Definition 3.9. Let V be a vector space, $n \geq 1$, and $r, s \in \mathbb{N}$. Suppose that $f : V \times V^r \to V$, $g : V \times V^s \to V$ are smooth maps,

$\mathfrak{m} = [\mathfrak{m}^1, \ldots, \mathfrak{m}^n] \in M(r, n; \mathbb{Z})$, $\mathfrak{n} = [\mathfrak{n}^1, \ldots, \mathfrak{n}^n] \in M(s, n; \mathbb{Z})$ are connection matrices and $L \in M(r, s+1; \mathbb{Q})$. We say f is $(L, \mathfrak{m}, \mathfrak{n})$-*input dominated by* g, written $f <^{\imath}_{(L,\mathfrak{m},\mathfrak{n})} g$, if

(1) For all $(x_0, x_1, \cdots, x_s) \in V \times V^s$, we have

$$g(x_0; x_1, \cdots, x_s) = f(x_0; L(x_0, \cdots, x_s)).$$

(2) For $j \in \mathbf{n}$, we have $g(x_j; x_{\mathfrak{n}^j_1}, \cdots, x_{\mathfrak{n}^j_s}) = f(x_j; x_{\mathfrak{m}^j_1}, \cdots, x_{\mathfrak{m}^j_r})$.

Remarks 3.10. (1) If f is $(L, \mathfrak{m}, \mathfrak{n})$-input dominated by g then

$$L(x_j, x_{\mathfrak{n}^j_1}, \cdots, x_{\mathfrak{n}^j_s}) = (x_{\mathfrak{m}^j_1}, \cdots, x_{\mathfrak{m}^j_r}),\ j \in \mathbf{n},$$

and so a *necessary* condition for input domination is $\mathfrak{m}^j \subseteq \{j\} \cup \mathfrak{n}^j$, $j \in \mathbf{n}$. That is

$$\{\mathfrak{m}^j_1, \ldots, \mathfrak{m}^j_r\} \subset \{j, \mathfrak{n}^j_1, \ldots, \mathfrak{n}^j_s\},\ j \in \mathbf{n}.$$

(2) f is *input dominated* by g if we can find L so that $f <^{\imath}_{(L,\mathfrak{m},\mathfrak{n})} g$.
(3) If we can choose $L \in M(r, s+1; \mathbb{Z})$ so that $f <^{\imath}_{(L,\mathfrak{m},\mathfrak{n})} g$, we write $f <^{\imath,\mathbb{Z}}_{(L,\mathfrak{m},\mathfrak{n})} g$.

Definition 3.11. Let $\mathcal{M}$ and $\mathcal{N}$ be coupled identical cell networks with asymmetric inputs such that

(a) $n(\mathcal{M}) = n(\mathcal{N}) = n$.
(b) Cells in $\mathcal{M}$ have r inputs, cells in $\mathcal{N}$ have s inputs.
(c) If we fix an ordering $\mathbf{C}_1, \ldots, \mathbf{C}_n$ of the cells in $\mathcal{N}$, then the associated connection matrix is $\mathfrak{n} = [\mathfrak{n}^1, \ldots, \mathfrak{n}^n]$.

We say $\mathcal{M}$ is *input dominated* by $\mathcal{N}$, denoted $\mathcal{M} \prec_I \mathcal{N}$, if there exist $L \in M(r, s+1; \mathbb{Q})$ and an ordering of the cells of $\mathcal{M}$, with associated connection matrix $\mathfrak{m}$, such that for every $\mathcal{F} \in \mathcal{M}(\mathbb{L})$ there exists $\mathcal{G} \in \mathcal{N}(\mathbb{L})$ for which $f_{\mathcal{F}} <^{\imath}_{(L,\mathfrak{m},\mathfrak{n})} g_{\mathcal{G}}$. If $\mathcal{N} \prec_I \mathcal{M}$ and $\mathcal{M} \prec_I \mathcal{N}$, we say $\mathcal{M}$ and $\mathcal{N}$ are *input equivalent* and write $\mathcal{M} \sim_I \mathcal{N}$.

Remarks 3.12. (1) We write $\mathcal{M} \prec_{I,\mathbb{Z}} \mathcal{N}$ if $\mathcal{M} \prec_I \mathcal{N}$ and we can require the map L of the definition to lie in $M(r, s+1; \mathbb{Z})$. We similarly define $\mathcal{M} \sim_{I,\mathbb{Z}} \mathcal{N}$. In example 3.8 we have $\mathcal{M} \prec_{I,\mathbb{Z}} \mathcal{N}$ (indeed, $\mathcal{M} \sim_{I,\mathbb{Z}} \mathcal{N}$).
(2) Input equivalence and domination may be defined for networks with symmetric inputs. We refer to [1] for details.

Lemma 3.13. *With the notation and assumptions of definition 3.11, in particular* asymmetric *inputs, we have*

$$\mathbf{A}(\mathcal{M}) \subset \mathbf{A}(\mathcal{N}) \text{ iff } \mathcal{M} \prec_I \mathcal{N}.$$

Proof. If $\mathcal{M} \prec_I \mathcal{N}$, we have $\mathcal{M}(\mathbb{L}) \prec_L \mathcal{N}(\mathbb{L})$ and so $\mathbf{A}(\mathcal{M}) \subset \mathbf{A}(\mathcal{N})$ by theorem 3.4 (or direct verification). Conversely, let $M_0 = I, \ldots, M_r$ and $N_0 = I, \ldots, N_s$ denote the adjacency matrices for $\mathcal{M}$ and $\mathcal{N}$ respectively. Since $\mathbf{A}(\mathcal{M}) \subset \mathbf{A}(\mathcal{N})$, there exists $L = [d_\ell^q] \in M(r, s+1; \mathbb{Q})$ such that

$$M_\ell = \sum_{q=0}^{s} d_\ell^q N_q,\ \ell \in \mathbf{r},$$

Suppose that $\mathcal{F} \in \mathcal{M}(\mathbb{L})$ has model $f : V \times V^r \to V$. Define $g : V \times V^s \to V$ by

$$g(x_0; x_1, \cdots, x_s) = f(x_0; L(x_0, \cdots, x_s)).$$

In order to prove that $f <^{\imath}_{(L,\mathfrak{m},\mathfrak{n})} g$, it suffices to verify that

$$L(x_j; x_{\mathfrak{n}_1^j}, \cdots, x_{\mathfrak{n}_s^j}) = (x_{\mathfrak{m}_1^j}, \cdots, x_{\mathfrak{m}_r^j}),\ j \in \mathbf{n}. \tag{3.3}$$

Let $j \in \mathbf{n}$. We have

$$m_{ij}^\ell = \sum_{q=0}^{s} d_\ell^q n_{ij}^q,\ i \in \mathbf{n}.$$

Multiply this equation by x_i and sum over i to obtain

$$\begin{aligned} \sum_{i=1}^{n} m_{ij}^\ell x_i &= \sum_{i=1}^{n} \sum_{q=0}^{s} d_\ell^q n_{ij}^q x_i \\ &= d_\ell^0 \sum_{i=1}^{n} n_{ij}^0 x_i + \sum_{q=1}^{s} d_\ell^q \sum_{i=1}^{n} n_{ij}^q x_i \end{aligned}$$

By (2.1), we have $\sum_{i=1}^n m_{ij}^\ell x_i = x_{\mathfrak{m}_\ell^j}$, $\ell \in \mathbf{r}$, $\sum_{i=1}^n n_{ij}^0 x_i = x_j$, and $\sum_{i=1}^n n_{ij}^q x_i = x_{\mathfrak{n}_q^j}$, $q \in \mathbf{s}$. Hence

$$x_{\mathfrak{m}_\ell^j} = d_\ell^0 x_j + \sum_{q=1}^{s} d_\ell^q x_{\mathfrak{n}_q^j},\ \ell \in \mathbf{r}.$$

Since the ℓth component of $L(x_j; x_{\mathfrak{n}_1^j}, \cdots, x_{\mathfrak{n}_s^j})$ is $d_\ell^0 x_j + \sum_{q=1}^s d_\ell^q x_{\mathfrak{n}_q^j}$, we have proved (3.3) and so $\mathcal{M} \prec_I \mathcal{N}$. □

Remark 3.14. Using the same proof, lemma 3.13 holds for scalar signalling networks $\mathcal{M}(\mathbb{S})$ (cf. example 2.12). If we deny self loops (see remarks 2.13(2)), then we work with the subspaces of $M(n; \mathbb{Q})$ generated by the non-identity adjacency matrices.

Proposition 3.15. *(Notation of definitions 3.11, 2.11.)* $\mathcal{N} \sim_I \mathcal{M}$ *iff* $\mathcal{N}(\mathbb{L}) \sim \mathcal{M}(\mathbb{L})$ *iff* $\mathcal{N}(\mathbb{S}) \sim \mathcal{M}(\mathbb{S})$.

Proof. If $\mathcal{N} \sim_I \mathcal{M}$ then obviously $\mathcal{N}(\mathbb{S}) \sim \mathcal{M}(\mathbb{S})$ and $\mathcal{N}(\mathbb{L}) \sim \mathcal{M}(\mathbb{L})$. Conversely, if either $\mathcal{N}(\mathbb{S}) \sim \mathcal{M}(\mathbb{S})$ or $\mathcal{N}(\mathbb{L}) \sim \mathcal{M}(\mathbb{L})$, then we have $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ (theorem 3.4) and so $\mathcal{N} \sim_I \mathcal{M}$ by lemma 3.13. □

Remark 3.16. Lemma 3.13 and proposition 3.15 in general *fail* for networks which have (some) symmetric inputs — see example 3.17 and [1]. However, there is no difficulty in extending the results to networks which have more than one class of cell as long as inputs are asymmetric. In particular, the algebraic condition $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ is required to hold for each cell class.

We conclude with a simple example that shows that if we allow symmetric inputs then equivalent systems need not be input equivalent.

Example 3.17. Suppose $p = q = 1$. Let $N_1 = \begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}$, $M_1 = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$. We have $M_1 = N_1 - I$ and so $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$. If $\mathcal{M} \prec_I \mathcal{N}$ then the input dominance relation is given by

$$g(x_0; x_1, x_2, x_3) = f(x_0; ax_0 + b(x_1 + x_2 + x_3), cx_0 + d(x_1 + x_2 + x_3))$$

for some $a, b, c, d \in \mathbb{R}$. This is the only possible relation since g has to be symmetric in the last three coordinates (the inputs are symmetric). Since $y' = g(y; x, x, y) = f(y; x, x)$, $ay + b(2x + y) = x, cy + d(2x + y) = x$ therefore, $a = c = -\frac{1}{2}, b = d = \frac{1}{2}$. It is easy to check that $x' = g(x; x, x, y) \neq f(x; x, y)$. Therefore $\mathcal{M} \not\prec_I \mathcal{N}$ and so $\mathcal{M}$, $\mathcal{N}$ are dynamically equivalent networks that are not input equivalent. More subtle examples appear in [1].

3.2. **Output equivalence.** Output equivalence is applicable to coupled cell networks with general phase space and continuous model as well as some classes of discrete system. The basic idea is that a network $\mathcal{N}$ 'output dominates' a network $\mathcal{M}$ if given a system $\mathcal{F} \in \mathcal{M}$, we can find a system in $\mathcal{G} \in \mathcal{N}$ which has identical dynamics to $\mathcal{F}$ and is such that each cell in $\mathcal{G}$ is built from several cells in $\mathcal{F}$ together with a single Add-Subtract cell and scaling cells acting on (un-integrated) outputs (see figure 5(b)). We start with a simple example to illustrate the ideas.

Example 3.18. Let $\mathcal{M}, \mathcal{N}$ be the network architectures of example 3.8 (see figure 6) We show $\mathcal{M} \prec \mathcal{N}$ by combining outputs rather than inputs. Suppose that the model for $\mathcal{F} \in \mathcal{M}$ is

$$\begin{aligned} x_1' &= f(x_1; x_1, x_2), \\ x_2' &= f(x_2; x_1, x_2). \end{aligned}$$

We look for a model g for $\mathcal{G} \in \mathcal{N}$, $\mathcal{F} \prec \mathcal{G}$, such that

$$g(x_0; x_1, x_2) = \sum_{\gamma} c_\gamma f_\gamma(x_0; x_1, x_2),$$

where the sum is over all maps $\gamma : \{1, 2\} \to \{0, 1, 2\}$, $f_\gamma(x_0; x_1, x_2) = f(x_0; x_{\gamma(1)}, x_{\gamma(2)})$, and $c_\gamma \in \mathbb{Q}$. In order that the dynamics of $\mathcal{G}$ is identical to that of $\mathcal{F}$ it suffices that $g(x; x, y) = f(x; x, y)$ and $g(x; y, y) = f(x; y, x)$. A straightforward computation shows that there is a two parameter family of solutions for g given by

$$\begin{aligned} g(x_0; x_1, x_2) \quad = \quad & \alpha f(x_0; x_1, x_2) + (\alpha - 1) f(x_0; x_0, x_1) + \\ & (1 - \alpha) f(x_0; x_0, x_2) + (1 - \beta) f(x_0; x_1, x_0) + \\ & (\beta - \alpha) f(x_0; x_1, x_1) + \beta f(x_0; x_2, x_0) - \beta f(x_0; x_2, x_1), \end{aligned}$$

where $\alpha, \beta \in \mathbb{Q}$. If we take $\alpha = 1$, $\beta = 0$, we get

$$g(x_0; x_1, x_2) = f(x_0; x_1, x_2) + f(x_0; x_1, x_0) - f(x_0; x_1, x_1)$$

which should be compared with the solution found in example 3.8. We can build the system $\mathcal{G}$ so as to realize the dynamics of $\mathcal{F}$ using Add-Subtract cells together with the cell used in $\mathcal{F}$. See figure 8 (and also figure 5(b)). Observe that the system shown in figure 8 can be

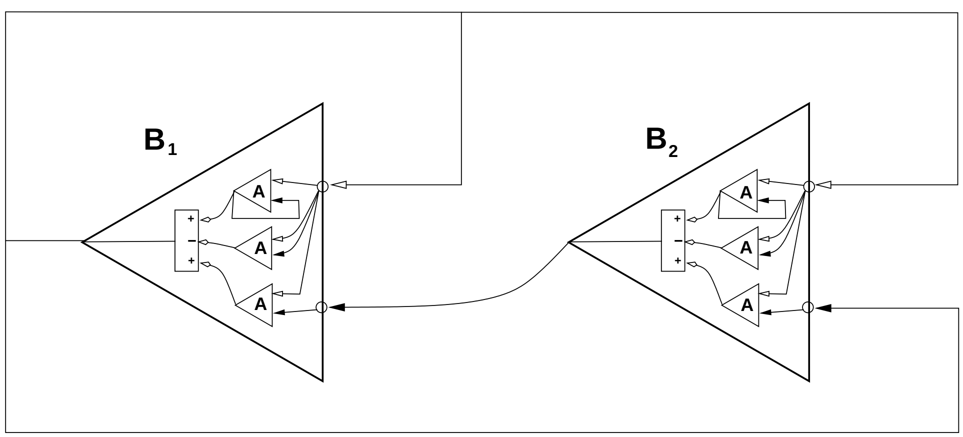


FIGURE 8. The system $\mathcal{G}$ built using 3 cells from $\mathcal{F}$ and an Add-Subtract cell.

transformed back into the system $\mathcal{F}$ by removing the outer triangles defining cells of class **B** and then cancelling outputs using linearity (of vector fields). Note that unlike the input based analysis of example 3.8, this configuration works whatever the phase space.

We now formalize the concepts of output domination and equivalence. For simplicity, we work with the case of asymmetric inputs.

However, the definitions extend easily and transparently to allow for symmetric inputs (for details see [1]).

Suppose that M is a smooth manifold and $f : M \times M^r \to TM$, $g : M \times M^s \to TM$ are smooth families of vector fields on M, $r, s \in \mathbb{Z}^+$. Let $\mathbf{A}(r,s)$ denote the set of all maps $\gamma : \{1,\dots,r\} \to \{0,\dots,s\}$. If $\gamma \in \mathbf{A}(r,s)$, define $f_\gamma : M \times M^s \to TM$ by

$$f_\gamma(x_0; x_1,\dots,x_s) = f(x_0; x_{\gamma(1)},\dots,x_{\gamma(r)}),\ (x_0,(x_1,\dots,x_s)) \in M \times M^s).$$

(Addition is in $T_{x_0}M$.)

Definition 3.19. Let M be a vector space, $n \geq 1$, and r, s be non-negative integers. Suppose that $f : M \times M^r \to TM$, $g : M \times M^s \to TM$ are families of vector fields, $\mathfrak{m} = [\mathfrak{m}^1,\dots,\mathfrak{m}^n] \in M(r,n;\mathbb{Z})$, $\mathfrak{n} = [\mathfrak{n}^1,\dots,\mathfrak{n}^n] \in M(s,n;\mathbb{Z})$ are connection matrices and $C : \mathbf{A}(r,s) \to \mathbb{Q}$. We say f is $(C,\mathfrak{m},\mathfrak{n})$-*output dominated by* g, written $f <^o_{(C,\mathfrak{m},\mathfrak{n})} g$, if

1. $g = \sum_{\gamma \in \mathbf{A}(r,s)} C(\gamma) f_\gamma$ (as a sum of vector fields on M).
2. For $j \in \mathbf{n}$ we have $g(x_j, x_{\mathfrak{n}^j_1},\cdots,x_{\mathfrak{n}^j_s}) = f(x_j; x_{\mathfrak{m}^j_1},\cdots,x_{\mathfrak{m}^j_r})$.

Remarks 3.20. (1) Just as for input domination, a *necessary* condition for output domination is

$$\{\mathfrak{m}^j_1,\dots,\mathfrak{m}^j_r\} \subset \{j,\mathfrak{n}^j_1,\dots,\mathfrak{n}^j_s\},\ j \in \mathbf{n}.$$

(2) We say f is *output dominated* by g if $f <^o_{(C,\mathfrak{m},\mathfrak{n})} g$ for some choice of C.
(3) It is possible to extend definition 3.19 to apply to discrete systems defined on compact M provided that $f : M \times M^r \to M$ is sufficiently C^0-close to the projection $\pi(x_0; x_1,\dots,x_r) = x_0$. For this, we may use the exponential map of a Riemannian metric on M so as to define uniform local linear coordinate systems at every point of M.

Definition 3.21. Let $\mathcal{M}$ and $\mathcal{N}$ be coupled identical cell networks such that

(a) $n(\mathcal{M}) = n(\mathcal{N}) = n$.
(b) Cells in $\mathcal{M}$ have r inputs, cells in $\mathcal{N}$ have s inputs.
(c) If we fix an ordering $\mathbf{C}_1,\dots,\mathbf{C}_n$ of the cells in $\mathcal{N}$, then the associated connection matrix is $\mathfrak{n} = [\mathfrak{n}^1,\dots,\mathfrak{n}^n]$.

We write $\mathcal{M} \prec_O \mathcal{N}$ if there exist $C : \mathbf{A}(r,s) \to \mathbb{Q}$ and an ordering of the cells of $\mathcal{M}$, with associated connection matrix $\mathfrak{m}$, such that for every $\mathcal{F} \in \mathcal{M}$, there exists $\mathcal{G} \in \mathcal{N}$ for which $f <^o_{(C,\mathfrak{m},\mathfrak{n})} g$. If $\mathcal{N} \prec_O \mathcal{M}$ and $\mathcal{M} \prec_O \mathcal{N}$, we say $\mathcal{N}$ and $\mathcal{M}$ are *output equivalent* and write $\mathcal{N} \sim_O \mathcal{M}$.

Remark 3.22. We write $\mathcal{M} \prec_{O,\mathbb{Z}} \mathcal{N}$ if $\mathcal{M} \prec_O \mathcal{N}$ and we can choose the map C of the definition to be $\mathbb{Z}$-valued. We similarly define $\mathcal{M} \sim_{O,\mathbb{Z}} \mathcal{N}$. We have $\mathcal{M} \prec_{O,\mathbb{Z}} \mathcal{N}$ in example 3.18 (indeed, $\mathcal{M} \sim_{O,\mathbb{Z}} \mathcal{N}$).

Theorem 3.23. *(Notation and assumptions as above.)* $\mathcal{N} \sim_O \mathcal{M}$ *iff* $\mathbf{A}(\mathcal{N}) = \mathbf{A}(\mathcal{M})$ *iff* $\mathcal{N} \sim \mathcal{M}$.

We prove theorem 3.23 when the cells have asymmetric inputs. The proof for the case of symmetric inputs is given in [1]. We break the proof of theorem 3.23 into a number of lemmas of independent interest. These lemmas also give a simple algorithm for computing an explicit output equivalence. As remarked in the introduction, the non-trivial part of this result is the construction of the output equivalence, granted that the algebraic condition $\mathbf{A}(\mathcal{N}) = \mathbf{A}(\mathcal{M})$ is satisfied.

Let $\mathcal{M}$ be a coupled n identical cell network and suppose that cells in $\mathcal{M}$ have $r \geq 1$ asymmetric inputs. and let $\mathbb{A}(\mathcal{M}) = \{M_0, M_1, \ldots, M_r\}$ be the set of adjacency matrices. Given $u \in \mathbf{r}$, let $\mathcal{M}^{-u}$ be the n identical cell network with $r-1$ asymmetric inputs and $\mathbb{A}(\mathcal{M}^{-u}) = \mathbb{A}(\mathcal{M}) \smallsetminus \{M_u\}$. That is, $\mathcal{M}^{-u}$ is obtained from $\mathcal{M}$ by removing the uth input from each cell.

Lemma 3.24. *(Notation and assumptions as above.) If* $u \in \mathbf{r}$ *then* $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{M}^{-u})$ *iff* $\mathcal{M} \sim_O \mathcal{M}^{-u}$.

Proof. We start by showing that $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{M}^{-u})$ implies $\mathcal{M} \sim_O \mathcal{M}^{-u}$. Permuting inputs we may and shall assume $u = r$. If $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{M}^{-r})$, then

$$M_r = \sum_{i=0}^{r-1} d^i M_i, \tag{3.4}$$

where $d^0, \ldots, d^{r-1} \in \mathbb{Q}$. It suffices to show $\mathcal{M} \prec_O \mathcal{M}^{-r}$ (the reverse order is trivial). Suppose $\mathcal{F} \in \mathcal{M}$ has model $f : M \times M^r \rightarrow TM$. Define $g : M \times M^{r-1} \rightarrow TM$ by

$$g(x_0; x_1, \cdots, x_{r-1}) = \sum_{i=0}^{r-1} d^i f(x_0; x_1, \cdots, x_{r-1}, x_i).$$

Using (3.4), we show easily that if $j \in \mathbf{n}$, then

$$g(x_j, x_{\mathfrak{m}_1^j}, \cdots, x_{\mathfrak{m}_{r-1}^j}) = f(x_j; x_{\mathfrak{m}_1^j}, \cdots, x_{\mathfrak{m}_r^j}), \tag{3.5}$$

where $[\mathfrak{m}^1, \ldots, \mathfrak{m}^n]$ is the connection matrix for $\mathcal{M}$. If $\mathcal{G} \in \mathcal{M}^{-r}$ has model g, then f is output dominated by g. Hence $\mathcal{M} \prec_O \mathcal{M}^{-r}$. It remains to show that if $\mathcal{M} \sim_O \mathcal{M}^{-u}$ then $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{M}^{-u})$. This can either be seen by reversing the previous argument or by observing that if $\mathcal{M} \sim_O \mathcal{M}^{-u}$ then certainly $\mathcal{M}(\mathbb{L}) \sim_L \mathcal{M}^{-u}(\mathbb{L})$ and so $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{M}^{-u})$ by theorem 3.4. $\square$

Lemma 3.25. *(Notation and assumptions as above.) If the network* $\mathcal{M}^\star$ *is derived from* $\mathcal{M}$ *by removing inputs so that*

(a) $\mathbf{A}(\mathcal{M}^\star) = \mathbf{A}(\mathcal{M})$,

(b) $\mathbb{A}(\mathcal{M}^\star)$ *is a linearly independent set (and so a basis for* $\mathbf{A}(\mathcal{M})$*),*
then $\mathcal{M}^\star \sim_O \mathcal{M}$.

Proof. The result follows by repeated application of lemma 3.24. □

Remark 3.26. For networks with *asymmetric* inputs, $\mathcal{M}^\star$ is automatically *minimal* in the sense of Aguiar & Dias [2]. That is, the number of inputs of $\mathcal{M}^\star$ is minimal. However, if we allow symmetric inputs then $\mathcal{M}^\star$ may not be minimal even if $\mathcal{M}^\star$ satisfies (a) and (b) of the lemma.

Lemma 3.27. *If* $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ *then* $\mathcal{M} \sim_O \mathcal{N}$.

Proof. It follows from lemma 3.25 that we can assume that $\mathbb{A}(\mathcal{M})$, $\mathbb{A}(\mathcal{N})$ both define bases of $\mathbf{A}(\mathcal{M})$. In particular, cells in both networks have the same number of inputs. Let $\mathbb{A}(\mathcal{M}) = \{M_0, \dots, M_r\}$, $\mathbb{A}(\mathcal{N}) = \{N_0, \dots, N_r\}$. Suppose that there is exactly one $j \in \mathbf{r}$ such that $N_j \neq M_j$ (of course, $M_0 = N_0 = I$). Permuting inputs, it is no loss of generality to assume $j = r$. We prove $\mathcal{M} \prec_O \mathcal{N}$. Since $M_r \in \mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$, We may write

$$M_r = \sum_{i=0}^{r} d^i N_i, \tag{3.6}$$

where the coefficients $d^i \in \mathbb{Q}$ and are unique. Suppose $\mathcal{F} \in \mathcal{M}$ has model $f : M \times M^r \to TM$. If we define $g : M \times M^r \to TM$ by

$$g(x_0; x_1, \cdots, x_r) = \sum_{i=0}^{r} d^i f(x_0; x_1, \cdots, x_{r-1}, x_i).$$

then g will be the model for a system $\mathcal{G} \in \mathcal{N}$ and g will output dominate f. The proof of the reverse order $\mathcal{N} \prec_O \mathcal{M}$ is exactly the same. The general case now follows by observing that we can transform $\mathcal{M}$ into $\mathcal{N}$ by modifying one input at a time. □

Remarks 3.28. (1) Lemmas 3.24, 3.27 give an iterative algorithm for constructing an explicit output equivalence. Even if $\mathbb{A}(\mathcal{M})$, $\mathbb{A}(\mathcal{N})$ are both bases, the output equivalence need not be unique — see example 3.18.

(2) Using similar methods to those given above, we can show that if $\mathbf{A}(\mathcal{M}) \subset \mathbf{A}(\mathcal{N})$ then $\mathcal{M} \prec_O \mathcal{N}$. Indeed, we may give an explicit formula that realizes the output domination. Suppose that cells in $\mathcal{M}$ have r inputs, cells in $\mathcal{N}$ have s inputs. For each $u \in \mathbf{r}$, let $M_u = \sum_{i=0}^{s} d_u^i N_i$ where $[d_u^i] \in M(r+1, s; \mathbb{Q})$. If $\mathcal{F} \in \mathcal{M}$ has model f, and we define

$$g(x_0; x_1, \cdots, x_s) = \sum_{i_1=0}^{s} \cdots \sum_{i_r=0}^{s} \left(\prod_{u=1}^{r} d_u^{i_u}\right) f(x_0; x_{i_1}, \cdots, x_{i_r}),$$

then g models the required system $\mathcal{G} \in \mathcal{N}$.

Lemma 3.29.

$$\mathcal{M} \sim_O \mathcal{N} \Longrightarrow \mathcal{M} \sim \mathcal{N} \Longrightarrow \mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N}).$$

Proof. If $\mathcal{M} \sim_O \mathcal{N}$ then obviously $\mathcal{M} \sim \mathcal{N}$. Hence, $\mathcal{M}(\mathbb{L}) \sim \mathcal{N}(\mathbb{L})$ and so $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ by theorem 3.4. □

Proof of Theorem 3.23 The proof is immediate from lemma 3.27 and lemma 3.29. □

Remarks 3.30. (1) Theorem 3.23 extends easily to networks containing more than one class of cell. Output equivalence holds iff we can index cells so that the condition $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ holds for each cell class.
(2) Theorem 3.23 applies to scalar signalling networks (definition 2.11) — the proof is formally identical.

4. Examples

4.1. **Systems with toral phase space.** In this section we consider coupled systems with phase space a torus $\mathbb{T}^q$, $q \geq 1$ (more generally, everything we say works for a phase space of the form $\mathbb{R}^p \times \mathbb{T}^q$, $q \geq 1$). Suppose that $\mathcal{M}$, $\mathcal{N}$ are coupled cell networks. It follows from theorem 3.23 that $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ iff $\mathcal{M} \sim_O \mathcal{N}$. So if $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$, we always have $\mathcal{M}(\mathbb{T}) \sim_O \mathcal{N}(\mathbb{T})$. As we shortly see, this is not necessarily so if we work in terms of input equivalence. That is, $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$ does *not* generally imply $\mathcal{M}(\mathbb{T}) \sim_I \mathcal{N}(\mathbb{T})$. However, if $\mathcal{M} \sim_{I,\mathbb{Z}} \mathcal{N}$, then we do have $\mathcal{M}(\mathbb{T}) \sim_I \mathcal{N}(\mathbb{T})$. This is so since adding integer multiples of angles gives a well-defined angle. Thus, the networks of example 3.8 are input equivalent.

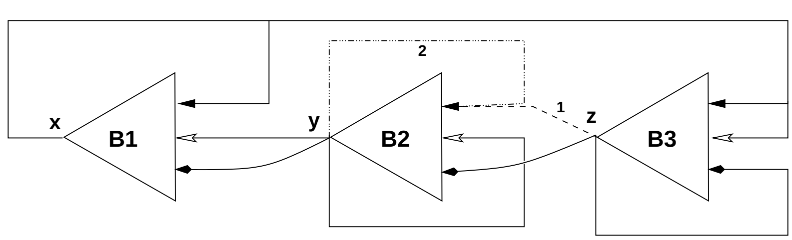


Figure 9

Example 4.1. In figure 9 we show two networks $\mathcal{M}$, $\mathcal{N}$ that differ only in the first input to $\mathbf{B}_2$. For the network $\mathcal{M}$, this input comes from $\mathbf{B}_3$, for $\mathcal{N}$ it comes from $\mathbf{B}_2$. The non-identity adjacency matrices for $\mathcal{M}$ are

$$M_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},\ M_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ M_3 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

The adjacency matrices for $\mathcal{N}$ are given by $N_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $N_2 = M_2, N_3 = M_3$. It is straightforward to verify that $\mathbb{A}(\mathcal{M})$, $\mathbb{A}(\mathcal{N})$ both define a basis for $\mathbf{A}$. Moreover,

$$\begin{aligned} N_1 &= \frac{1}{2}(M_0 + M_1 + M_2 - M_3), \\ M_1 &= -N_0 + 2N_1 - N_2 + N_3, \end{aligned}$$

and so $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N}) = \mathbf{A}$. Suppose that $\mathcal{F} \in \mathcal{M}(\mathbb{T})$ has model f and $\mathcal{G} \in \mathcal{N}(\mathbb{T})$ has model g. We assume both systems have phase space $\mathbb{T}$ and denote the variables for $\mathcal{F}$ by θ_i and for $\mathcal{G}$ by ϕ, $i \in \mathbf{3}$. With these conventions the differential equations for $\mathcal{F}$ and $\mathcal{G}$ are given by

$$\begin{aligned} \theta_1' &= f(\theta_1; \theta_1, \theta_2, \theta_2), & \phi_1' &= g(\phi_1; \phi_1, \phi_2, \phi_2), \\ \theta_2' &= f(\theta_2; \theta_3, \theta_2, \theta_3), & \phi_2' &= g(\phi_2; \phi_2, \phi_2, \phi_3), \\ \theta_3' &= f(\theta_3; \theta_1, \theta_1, \theta_3), & \phi_3' &= g(\phi_3; \phi_1, \phi_1, \phi_3), \end{aligned}$$

Since $\mathbf{A}(\mathcal{M}) = \mathbf{A}(\mathcal{N})$, we have $\mathcal{M} \sim \mathcal{N}$ and so $\mathcal{M} \sim_O \mathcal{N}$. In particular, if $\mathcal{F}$ and $\mathcal{G}$ are output equivalent then an output equivalence is given by

$$\begin{aligned} f(\theta_0; \theta_1, \theta_2, \theta_3) &= \frac{1}{2}(g(\theta_0; \theta_0, \theta_2, \theta_3) + g(\theta_0; \theta_1, \theta_2, \theta_3) \\ &\quad + g(\theta_0; \theta_2, \theta_2, \theta_3) - g(\theta_0; \theta_3, \theta_2, \theta_3)), \\ g(\phi_0; \phi_1, \phi_2, \phi_3) &= -f(\phi_0; \phi_0, \phi_2, \phi_3) + 2f(\phi_0; \phi_1, \phi_2, \phi_3) \\ &\quad - f(\phi_0; \phi_2, \phi_2, \phi_3) + f(\phi_0; \phi_3, \phi_2, \phi_3). \end{aligned}$$

We emphasize these relations are not unique. There is a system of 24 linear equations in 64 unknowns which determine the possible output equivalences, we present one solution from a 40-dimensional family. Our solution is given by the proof of theorem 3.23. The question of input equivalence is more subtle. We have $\mathcal{F} \prec_I \mathcal{G}$ if and only if

$$g(\phi_0; \phi_1, \phi_2, \phi_3) = f(\phi_0; 2\phi_1 - \phi_0 - \phi_2 + \phi_3, \phi_2, \phi_3). \tag{4.7}$$

The input equivalence is uniquely determined and well defined since coefficients are all integers. On the other hand, $\mathcal{G} \not\prec_I \mathcal{F}$ since the relation for input equivalence has to be

$$f(\theta_0;\theta_1,\theta_2,\theta_3)=g(\theta_0;\frac{1}{2}(\theta_0+\theta_1+\theta_2-\theta_3),\theta_2,\theta_3),$$

and this is not well-defined on the torus.

If instead we consider discrete dynamical systems on $\mathbb{T}$, we see that if (4.7) holds then $\mathcal{M}\prec_{I,\mathbb{Z}}\mathcal{N}$ and $\mathcal{M}\prec_{O,\mathbb{Z}}\mathcal{N}$. Hence $\mathcal{M}(\mathbb{T})\prec\mathcal{N}(\mathbb{T})$ for discrete dynamics. The converse relation is less clear. Certainly, $\mathcal{N}\not\prec_{I,\mathbb{Z}}\mathcal{M}$. It is conceivable that $\mathcal{N}\prec_{O,\mathbb{Z}}\mathcal{M}$ if there exist output equivalences in the 40-dimensional family of solutions which have integer coefficients — however, it is easy to see that there are no such solutions. Nevertheless, there remains the possibility that $\mathcal{N}(\mathbb{T})\prec\mathcal{M}(\mathbb{T})$ for discrete dynamics. Indeed, if we assume that there exist $p\in\mathbb{Z}$ and $c\in\mathbb{T}$ such that $|g(\theta_0;\theta_1,\theta_2,\theta_3)-(c+p\theta_0)|<\pi/2$, for all $\theta_0,\ldots,\theta_3\in\mathbb{T}$, then, using the exponential map for $\mathbb{T}^3$, we can define f as above so that g is output dominated by f. More generally, we can always continuously deform f,g to their linearizations and reduce the question to a problem of output equivalence of discrete linear systems. For example, if we take $g(\phi_0;\phi_1,\phi_2,\phi_3)=\phi_0+\phi_1$, then it is not possible to find f realizing the same discrete dynamics as g. On the other hand if we take $g(\phi_0;\phi_1,\phi_2,\phi_3)=2(\phi_0+\phi_1)$, then we can find f realizing the same discrete dynamics. All of this shows that there are topological obstructions to the equivalence of $\mathcal{M}(\mathbb{T})$ and $\mathcal{N}(\mathbb{T})$ for discrete dynamics. None of these issues arise if we assume scalar signalling networks.

4.2. **Systems with non-Abelian group as a phase space.** We look at two examples where the phase space is the non-Abelian Lie group SO(3) (what we say holds for any connected non-Abelian Lie group).

Example 4.2. We consider the networks $\mathcal{M}$ and $\mathcal{N}$ of example 3.8. We choose systems $\mathcal{F}\in\mathcal{M}(\mathrm{SO}(3))$ and $\mathcal{G}\in\mathcal{N}(\mathrm{SO}(3))$. Denote the corresponding models by f and g respectively where $f,g:\mathrm{SO}(3)\times\mathrm{SO}(3)^2\to T\mathrm{SO}(3)$. In this case we may define input equivalence using the group structure on SO(3). Specifically, if we have

$$g(\gamma_0;\gamma_1,\gamma_2)=f(\gamma_0;\gamma_1,\gamma_0\gamma_1^{-1}\gamma_2),\ \gamma_0,\gamma_1,\gamma_2\in\mathrm{SO}(3),$$

then

$$\begin{aligned}g(\gamma_1;\gamma_1,\gamma_2)&=f(\gamma_1;\gamma_1,\gamma_1\gamma_1^{-1}\gamma_2)=f(\gamma_1;\gamma_1,\gamma_2),\\ g(\gamma_2;\gamma_1,\gamma_1)&=f(\gamma_2;\gamma_1,\gamma_2\gamma_1^{-1}\gamma_1)=f(\gamma_2;\gamma_1,\gamma_2),\end{aligned}$$

and so f is input dominated by g (note that the order of the composition $\gamma_1\gamma_1^{-1}\gamma_2$ matters). We obtain the reverse relation by taking

$$g(\gamma_0;\gamma_1,\gamma_2)=f(\gamma_0;\gamma_1,\gamma_1\gamma_0^{-1}\gamma_2),\ \gamma_0,\gamma_1,\gamma_2\in\mathrm{SO}(3).$$

Hence $\mathcal{M}(\mathrm{SO}(3)) \sim_I \mathcal{N}(\mathrm{SO}(3))$. Exactly the same arguments show that for discrete dynamics on SO(3) we have both input and output equivalence with these network structures.

Example 4.3. We start by considering input equivalence for the networks $\mathcal{M}, \mathcal{N}$ of example 4.1 when the phase space is SO(3) and we consider continuous dynamics. Suppose that $\mathcal{F} \in \mathcal{M}(\mathrm{SO}(3))$ has model f, where $f : \mathrm{SO}(3) \times \mathrm{SO}(3)^3 \to T\mathrm{SO}(3)$. We attempt to construct a model g for $\mathcal{G} \in \mathcal{N}$ which input dominates f. For example, we can try $g(\gamma_0; \gamma_1, \gamma_2, \gamma_3) = f(\gamma_0; \gamma_2^{-1}\gamma_3\gamma_0^{-1}\gamma_1^2, \gamma_2, \gamma_3)$. We find that inputs do not match for the second cell:

$$g(\phi_2; \phi_2, \phi_2, \phi_3) = f(\phi_2; \phi_2^{-1}\phi_3\phi_2, \phi_2, \phi_3) \neq f(\phi_2; \phi_3, \phi_2, \phi_3).$$

It is easy to verify that whatever the order of composition of γ_0^{-1}, γ_1^2, γ_2^{-1}, γ_3, inputs do not match for at least one cell. Consequently, $\mathcal{M}(\mathrm{SO}(3)) \not\prec_I \mathcal{N}(\mathrm{SO}(3))$.

Similar arguments show that $\mathcal{N}(\mathrm{SO}(3)) \not\prec_I \mathcal{M}(\mathrm{SO}(3))$ and that input domination either way fails for discrete dynamics. It is not clear whether or not we have $\mathcal{N}(\mathrm{SO}(3)) \prec_O \mathcal{M}(\mathrm{SO}(3))$ for discrete dynamics though the output equivalence that works for vector fields will not work for discrete dynamics. There is nothing we can say concerning the reverse relation. In particular, for discrete dynamics, we do not know whether either of the relations $\mathcal{N}(\mathrm{SO}(3)) \prec \mathcal{M}(\mathrm{SO}(3))$, $\mathcal{M}(\mathrm{SO}(3)) \prec \mathcal{N}(\mathrm{SO}(3))$ holds let alone whether or not we have equivalence.

4.3. **An example with symmetric inputs.** We conclude with a simple example consisting of a pair of two identical cell networks with three symmetric inputs.

Example 4.4. In figure 10, we show two equivalent networks $\mathcal{S}_1, \mathcal{S}_2$ with symmetric inputs. If $\mathcal{S}_1, \mathcal{S}_2$ have asymmetric inputs, it is easy to

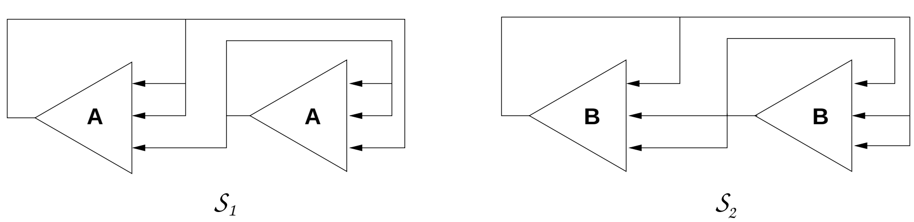


FIGURE 10. Two equivalent networks with symmetric inputs

see that they are equivalent and that the equivalence may be realized by either input or output equivalence. If we denote the adjacency matrices of $\mathcal{S}_i$ by $M_0^i = I$ and M_1^i, then $M_1^2 = 2M_1^1 - 3M_0^1$, $M_1^1 = \frac{1}{2}(M_1^2 + 3M_0^2)$ and so $\mathbf{A}(\mathcal{S}_1) = \mathbf{A}(\mathcal{S}_2)$ and the networks are equivalent by the (symmetric input version of) theorem 3.23. It is not hard to verify that $\mathcal{S}_1$ and $\mathcal{S}_2$ are input equivalent. If we assume a system in $\mathcal{S}_1$ is modelled by f and a equivalent system in $\mathcal{S}_2$ is modelled by g then an explicit output equivalence is given by

$$\begin{aligned} g(x_0;\overline{x_1,x_2,x_3}) &= \frac{1}{2}\left(\sum_{1\le i<j\le 3} f(x_0;\overline{x_0,x_i,x_j}) - f(x_0;\overline{x_1,x_2,x_3})\right), \\ f(x_0;\overline{x_1,x_2,x_3}) &= -\sum_{i=1}^{3} g(x_0;\overline{x_0,x_0,x_i}) + \sum_{i=1}^{3} g(x_0;\overline{x_0,x_i,x_i}) \\ &\quad + g(x_0;\overline{x_1,x_2,x_3}). \end{aligned}$$

If we are given the **A** cells, we can realize the same dynamics in the network architecture $\mathcal{S}_2$ using the configuration shown in figure 11.

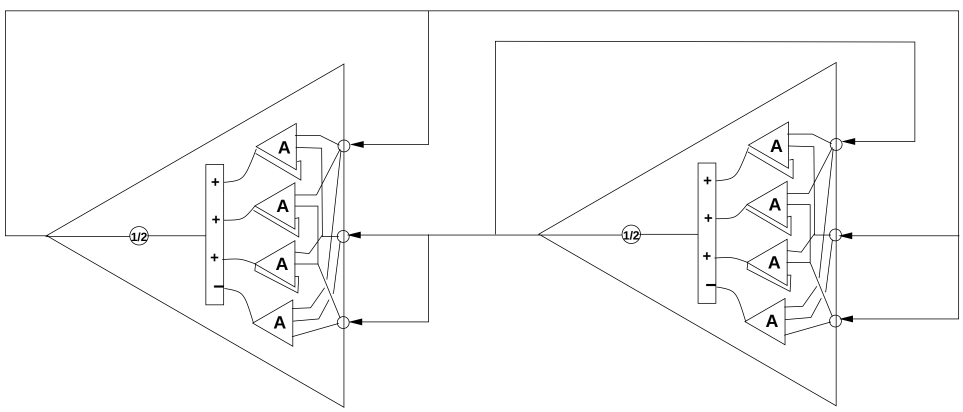


FIGURE 11. Realizing the dynamics of an $\mathcal{S}_1$ system using $\mathcal{S}_2$ architecture

Acknowledgments
Our research benefited greatly from fruitful conversations with Ana Dias and Manuela Aguiar during their visit to the University of Houston in 2009. We also wish to express our thanks to the referees who made many helpful suggestions to improve the clarity of the exposition.

REFERENCES

[1] N Agarwal and M Field. 'Dynamical equivalence of networks of coupled dynamical systems II: symmetric inputs', preprint.

[2] M Aguiar and A P S Dias. 'Minimal coupled cell networks', *Nonlinearity* **20** (2007), 193–219.

[3] M Aguiar, A P S Dias, P Ashwin, and M Field. 'Dynamics of coupled cell networks: synchrony, heteroclinic cycles and inflation', preprint, 2009.

[4] P Ashwin and J Swift. 'The dynamics of n weakly coupled identical oscillators', *Journal of Nonlinear Science* **2** (1992), 69–108.

[5] A P S Dias and I Stewart. 'Linear equivalence and ODE-equivalence for coupled cell networks', *Nonlinearity* **18** (2005), 1003–1020.

[6] M J Field. 'Combinatorial Dynamics', *Dynamical Systems* **19** (2004), 217–243.

[7] M Golubitsky, M Pivato and I Stewart. 'Interior symmetry and local bifurcation in coupled cell networks'. *Dynamical Systems* **19** (2004), 389–407.

[8] M Golubitsky and I Stewart. 'Nonlinear dynamics of networks: the groupoid formalism'. *Bull. Amer. Math. Soc.* **43** (2006), 305–364.

[9] M Golubitsky, I Stewart and A Török. 'Patterns of Synchrony in Coupled Cell Networks with Multiple Arrows', *SIAM J. Appl. Dynam. Sys.* **4**(1) (2005), 78–100.

[10] N Kashtan & U Alon. Spontaneous evolution of modularity and network motifs, *PNAS* **102** (39) (2005), 13773–13778.

[11] Y Kuramoto. *Chemical Oscillations, Waves and Turbulence*, (Springer-Verlag, Berlin, 1984).

[12] M E J Newman. 'The structure and function of complex networks', *SIAM Review* **45** (2003), 167–256.

[13] J G Restrepo, E Ott and B R Hunt. 'Emergence of synchronization in complex networks of interacting dynamical systems', *Physica D* **224** (2006), 114–122.

[14] I Stewart, M Golubitsky and M Pivato. 'Symmetry groupoids and patterns of synchrony in coupled cell networks', *SIAM J. Appl. Dynam. Sys.* **2** (4) (2003), 606–646.

Nikita Agarwal, Department of Mathematics, University of Houston, Houston, TX 77204-3008, USA

E-mail address: nagarwal@math.uh.edu

Michael Field, Department of Mathematics, University of Houston, Houston, TX 77204-3008, USA

E-mail address: mf@uh.edu