

# On Definition and Inference of Nonlinear Boolean Dynamic Networks

Zuogong Yue<sup>1\*</sup>, Johan Thunberg<sup>1</sup>, Lennart Ljung<sup>2</sup> and Jorge Gonçalves<sup>1</sup>

**Abstract**—Network reconstruction has become particularly important in systems biology, and is now expected to deliver information on causality. Systems in nature are inherently nonlinear. However, for nonlinear dynamical systems with hidden states, how to give a useful definition of dynamic networks is still an open question. This paper presents a useful definition of Boolean dynamic networks for a large class of nonlinear systems. Moreover, a robust inference method is provided. The well-known Millar-10 model in systems biology is used as a numerical example, which provides the ground truth of causal networks for key mRNAs involved in eukaryotic circadian clocks. In addition, as second contribution of this paper, we suggest definitions of linear network identifiability, which helps to unify the available work on network identifiability.

## I. INTRODUCTION

Network reconstruction is becoming increasingly important in systems biomedicine. With time series data available via gene expression microarrays, RNA sequencing, etc., it is expected that network reconstruction methods will help to deliver information about causality relationships, e.g. the identification of critical genes or pathways that are responsible for diseases. However, there are still many hurdles on the path of establishing such relationships, e.g. large systems dimensions, nonlinearity, low sampling rates, etc.

In regard to causality, *Granger causality* has to be mentioned. It is known that it can be used for nonlinear systems, however, it is not clear that what class of nonlinear systems it can be effectively applied to. Enlightened by the equivalence (under certain mild conditions [1]) between Granger causality and *vector autoregression* models [2], the systematic approach employed in the systems control community provides a way to revisit the Granger causality digraphs and extend their range of applications to a broader class of dynamical systems. The *dynamical structure function* (DSF) is an example [3], so is the similar model introduced in [4]. Both of them generalize the network representation to cover a larger class of the Linear Time-Invariant (LTI) systems. Moreover, system identification provides more efficient methods to perform network reconstructions (e.g. [5], [6]) than combinatorial pair-wise statistical tests.

For nonlinear dynamical systems there is a lack of powerful network representations/definitions that are equivalent

or similar to the DSF-like ones for LTI systems<sup>1</sup> to react to demands in biological applications. For the case of full-state measurements, i.e. when the entire state vector is measured, the definition of Boolean networks is straightforward via nonlinear state-space representations. This perspective has been used in [8], [9]. However, in the presence of *hidden states*, i.e. states that are not directly measured, the problem becomes particularly hard to tackle. This paper aims to give a useful definition of Boolean dynamic networks for nonlinear dynamical systems, which motivates us to propose an inference method that requires no *a priori* knowledge of nonlinear basis functions, and uses output measurements.

This paper generalizes the DSF [3] to a large class of nonlinear dynamical systems allowing the existence of hidden states. The nonlinear systems are on the form

$$\dot{x} = F(x) + Bu + Re, \quad (1a)$$

$$y = [I \ 0]x, \quad (1b)$$

where the state variable  $x$  is defined on  $M$  that is an open subset of  $\mathbb{R}^n$ ,  $F$  is a smooth vector field defined on  $M$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $R \in \mathbb{R}^{n \times p}$ ,  $y$  is of dimension  $p$  ( $p \leq n$ ),  $u$  is the control input of dimension  $m$ , and  $e$  is  $p$ -variate unknown white noise with covariance  $\mathbb{E}[e(t)e^T(\tau)] = I\delta(t-\tau)$ . Here, by slight abuse of notation, we use the model of white noise commonly used in the field of engineering to simplify the DSF notation (2). To be strict, (1) should be described with respect to Brownian motion, and its meaning is assigned by stochastic integrals [10]. Let  $x \triangleq [y^T \ z^T]^T$  be the state variables in (1), where the elements in  $y$  are the output variables and those in  $z$  are the hidden states (or *latent variables*).

## II. DIVING INTO LINEAR DYNAMIC NETWORKS

### A. Review of linear network models

Considering a continuous-time LTI system, we have the following general form of DSF [3], [11]

$$y(t) = Q(q)y(t) + P(q)u(t) + H(q)e(t), \quad (2)$$

where

$$\begin{aligned} Q(q) &= [Q_{ij}(q)]_{p \times p}, & Q_{ii}(q) &= 0, \quad \forall i, \\ P(q) &= [P_{ij}(q)]_{p \times m}, & H(q) &= [H_{ii}(q)]_{p \times p}, \end{aligned}$$

<sup>1</sup>This concept may be understood by analogy to the DSF or Granger Causality digraphs for LTI systems. The network considered in this paper is required to deliver direct causality. For instance, if mixing direct and indirect causality (many of the articles on “nonlinear networks” actually study such cases), the causal network of mRNAs in the circadian clock model [7] would be a complete digraph, which is no longer useful.

This work was supported by Fonds National de la Recherche Luxembourg (Ref. AFR-9247977, AFR-8864515).

<sup>1</sup> Zuogong Yue, Johan Thunberg and Jorge Gonçalves are with Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, 7 Avenue des Hauts Fourneaux, 4362, Esch-sur-Alzette, Luxembourg.

<sup>2</sup> Lennart Ljung is with Department of Electrical Engineering, Linköping University, Linköping, SE-58183, Sweden.

\* For correspondence, [zuogong.yue@uni.lu](mailto:zuogong.yue@uni.lu)

and  $Q_{ij}(q), P_{ij}(q), H_{ii}(q)$  are single-input-single-output (SISO) transfer functions,  $q$  is the differential operator, i.e.  $qy(t) = dy(t)/dt$ . For simplicity, we also refer to  $(Q, P, H)$  as the DSF (2). Furthermore, the *linear dynamic network*  $\mathcal{N} = (\mathcal{G}, f)$  is defined from the DSF (2) according to [6, Def. 1], where  $f$  is the *capacity function* of  $\mathcal{N}$ , and  $\mathcal{G}$  the *underlying digraph*, which is also called *linear Boolean (dynamic) network* due to the legacy of terminology in biology.

Consider the network model (2), which yields its input-output form  $y(t) = (I - Q)^{-1}Pu(t) + (I - Q)^{-1}He(t)$ . Referring to [12, chap. 4], we derive the network predictor model as follows (see e.g. [6] or [13]).

**Definition 1.** A *network predictor model* of an LTI system is a stable filter  $W(q)$  that defines a predictor  $\hat{y}(t|t-1) = W(q)r(t)$ , where  $r(t) = [u(t)^T \ y(t)^T]^T$ ,  $W(q) = [W_u(q) \ W_y(q)]$ , and

$$\begin{aligned} W_u(q) &= H^{-1}(q)P(q), \\ W_y(q) &= H^{-1}[Q(q) + H(q) - I]. \end{aligned} \quad (3)$$

### B. Linear network identifiability

There have been several studies on linear dynamic network identifiability, e.g. [3], [11], [14], covering different aspects of this concept. However, no literature explicitly gives a strict mathematical definition. We felt obliged to unify the definition, which can summarize the available results in a common language.

**Definition 2.** A *network model set*  $\mathcal{M}^*$  and *network predictor model set*  $\mathcal{M}_{\text{Pred}}^*$  are defined as follows

$$\mathcal{M}^* = \left\{ (Q(q), P(q), H(q))_\iota : \iota \in \mathcal{I} \right\}, \quad (4a)$$

$$\mathcal{M}_{\text{Pred}}^* = \{W_\iota(q) : \iota \in \mathcal{I}\}, \quad (4b)$$

where  $\mathcal{M}_{\text{Pred}}^*$  is a set of  $W(q)$ 's defined in (3), and  $\mathcal{I}$  denotes the general index set, which is not necessarily finite nor countable.

**Definition 3.** A *network model structure*  $\mathcal{M}$  and a *network predictor model structure*  $\mathcal{M}_{\text{Pred}}$  are differentiable maps from a connected open subset  $D_{\mathcal{M}} \subseteq \mathbb{R}^d$  to model sets

$$\begin{aligned} \mathcal{M} : D_{\mathcal{M}} &\rightarrow \mathcal{M}^* \\ \theta &\mapsto (Q(q, \theta), P(q, \theta), H(q, \theta)), \end{aligned} \quad (5a)$$

$$\begin{aligned} \mathcal{M}_{\text{Pred}} : D_{\mathcal{M}} &\rightarrow \mathcal{M}_{\text{Pred}}^* \\ \theta &\mapsto W(q, \theta), \end{aligned} \quad (5b)$$

such that the gradient of the predictor model is stable (see [12, p. 108] for this concept), i.e. for any given  $z \in \mathbb{C}$ ,  $|z| \geq 1$ , the gradient of  $W(z, \theta)$  over  $\theta$  exists and is stable for  $\theta \in D_{\mathcal{M}}$ , where  $W(z, \theta) = [H^{-1}(z, \theta)P(z, \theta) \ H^{-1}(z, \theta)[Q(z, \theta) + H(z, \theta) - I]]$ .

Now we present the definitions of network identifiability. Due to space limitations, we omit the ‘‘local’’ versions of the definitions, which can be defined similarly as [12, p. 113].

**Definition 4** (network identifiability). A network model structure  $\mathcal{M}$  is *globally identifiable at*  $\theta^*$  ( $\theta^* \in D_{\mathcal{M}}$ ) if

$$\mathcal{M}_{\text{Pred}}(\theta) = \mathcal{M}_{\text{Pred}}(\theta^*), \theta \in D_{\mathcal{M}} \Rightarrow \theta = \theta^*. \quad (6)$$

Moreover,  $\mathcal{M}$  is *globally identifiable* if it is globally identifiable at almost all  $\theta^* \in D_{\mathcal{M}}$ , i.e. the set of  $\theta$  at which  $\mathcal{M}$  is not globally identifiable has Lebesgue measure zero.

To put other available perspectives on network identifiability [3], [14] in a unified language, we introduce two more definitions by modifying (6) in Definition 4 with the following two statements, respectively,

[S1] (network identifiability with known structure)

$$\mathcal{M}(\theta) = \mathcal{M}(\theta^*), \theta \in D_{\mathcal{M}} \Rightarrow \theta = \theta^*; \quad (7)$$

[S2] (unique factorization from predictor models)

$$\mathcal{M}_{\text{Pred}}(\theta) = \mathcal{M}_{\text{Pred}}(\theta^*), \theta \in D_{\mathcal{M}} \Rightarrow \mathcal{N}(\theta) = \mathcal{N}(\theta^*), \quad (8)$$

where  $\mathcal{N}(\theta)$  denotes such an  $\mathcal{N}$  that is determined from  $[Q(\theta), P(\theta), H(\theta)]$  by [6, Def. 1].

*Remark 1.* Note that  $\mathcal{N}(\theta) = \mathcal{N}(\theta^*)$  is equivalent to  $\mathcal{M}(\theta) = \mathcal{M}(\theta^*)$ . This implies [S2] and [S1] together are Definition 4. We use  $\mathcal{N}$  in (8) so as to make it share a unified notation with Definition 5 below.

Compared to the definition of input-output system identifiability [12, p. 112,114], one may notice that [S1] is obtained by replacing the transfer function model set<sup>2</sup> in [12, p. 114] by the network model set. The condition [S1] can be interpreted as a definition of networks identifiability with known structures (i.e. the Boolean networks are known). The difference between (7) and (6) has to be emphasized. The model structures  $\mathcal{M}_{\text{Pred}}$  and  $\mathcal{M}$  are not equivalent. It is due to the fact that we need additional conditions to guarantee the unique factorization of  $(Q(q, \theta), P(q, \theta), H(q, \theta))$  from  $W(q, \theta)$ , which is formularized in mathematics as [S2]. This is the essential challenge in network reconstruction and has been used as the ‘‘definition’’ of network identifiability in [3], [11], [14]. In most applications we are particularly interested in the conditions that guarantee the unique identification of  $\mathcal{N}$  from data (even if the exactly same  $\mathcal{N}$  can be determined by different parameters  $\theta$ 's). See [13, Sec. II] for more comments on the study of network identifiability in [3], which considers a fairly general model parametrization.

When considering *Boolean dynamic networks*, it is likely that multiple  $\theta$ 's lead to the same Boolean structure. However, the essence is to guarantee unique Boolean structures from data. Therefore, the definition follows by replacing the dynamic network  $\mathcal{N}$  in (8) with its digraph  $\mathcal{G}$ .

**Definition 5** (Boolean network identifiability). The Boolean structure  $\mathcal{G}$  of a dynamic network with model structure  $\mathcal{M}$  is *globally identifiable at*  $\theta^*$  ( $\theta^* \in D_{\mathcal{M}}$ ) if

$$\mathcal{M}_{\text{Pred}}(\theta) = \mathcal{M}_{\text{Pred}}(\theta^*), \theta \in D_{\mathcal{M}} \Rightarrow \mathcal{G}(\theta) = \mathcal{G}(\theta^*). \quad (9)$$

Moreover, the Boolean network is *globally identifiable* if it is globally identifiable at almost all  $\theta^* \in D_{\mathcal{M}}$ .

<sup>2</sup>Here we consider the model set as a family of  $(G(q), H(q))$ 's instead of  $W(q)$ 's in [12, p. 107], by noticing that they are actually equivalent for input-output models [12, p. 105].

### III. NONLINEAR BOOLEAN DYNAMIC NETWORKS

#### A. Definition

Consider the dynamical system (1), where  $M$  is an open subset of  $\mathbb{R}^n$  and  $F : M \rightarrow M$  a smooth map. For each  $x \in M$  we have the *differential of  $F$  at  $x$*  defined as a map  $dF_x : T_x M \rightarrow T_{F(x)} M$ , where  $T_x M$  is the *tangent space to  $M$  at  $x$*  and  $T_{F(x)} M$  is the tangent space of  $M$  at the point  $F(x)$ . Moreover, let  $TM$  denote the *tangent bundle of  $M$* . The notations comply with [15] and one may consult [15, chap. 3] for more details. The matrix of  $dF_x$  in terms of the standard coordinate bases can be computed, denoted by  $A_x$ , which is irrelevant to the specific choice of coordinate charts. In particular,  $M$  is an open subset of Euclidean spaces, thus  $A_x$  is the Jacobian matrix of  $F$  at  $x$ .

Now, at  $x \in M$ , we have the dynamical system locally defined by the linear state-space representation  $(A_x, B, R)$ . Following the standard definition of DSF for LTI systems (see [3]), we have the dynamic network model defined locally as  $(Q_x(q), P_x(q), H_x(q))$ . Therefore, at  $x \in M$ , we have a Boolean dynamic network  $\mathcal{G}_x^{\text{loc}}$  defined as a digraph from  $(Q_x(q), P_x(q), H_x(q))$  (see [6, Def. 1]). Now the definition of nonlinear Boolean dynamic network naturally follows

**Definition 6.** A nonlinear Boolean dynamic network  $\mathcal{G}$  of the nonlinear dynamic system (1) is defined as follows:

$$V(\mathcal{G}) := V(\mathcal{G}_x^{\text{loc}}), \forall x \in M, \quad E(\mathcal{G}) := \bigcup_{x \in M} E(\mathcal{G}_x^{\text{loc}}), \quad (10)$$

where  $V(\mathcal{G})$  and  $E(\mathcal{G})$  denote the vertex and edge sets of  $\mathcal{G}$ .

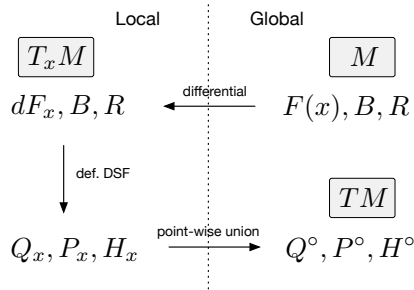


Fig. 1: A diagram illustrating the procedure of Definition 6. The nonlinear Boolean dynamic network  $\mathcal{G}$  can be equivalently denoted by  $(Q^o, P^o, H^o)$ , where  $Q^o := \bigvee_x Q_x^o, P^o := \bigvee_x P_x^o, H^o := \bigvee_x H_x^o$ ,  $\bigvee$  denotes the element-wise logical disjunction operator, and the logical matrix  $C^o$  of  $C(q)$  is defined as:  $C_{ij}^o = 0$  if  $C_{ij}(q) = 0$ , otherwise 1.

#### B. Additional Issues in Network Reconstruction

Definition 6 is based on  $M$ , which needs to be forward invariant under a specific class of input signals. However, during the process of network inference from data, only the outputs are accessible, which comprise a subset of the state trajectories. This issue, which indeed deserves more studies, is bypassed here by the following assumption and the, subsequently introduced, simplified definition of nonlinear network identifiability.

**Assumption 1.** The interconnection structure between state variables in dynamical systems is independent from the initial conditions and the inputs.

Let  $\mathcal{C}$  denote the signal of state variables when the dynamical system evolves through time. Assumption 1 implies that  $E(\mathcal{G}) = E(\mathcal{G}|_{\mathcal{C}})$ , where  $\mathcal{G}|_{\mathcal{C}}$  denote the Boolean network defined on  $\mathcal{C}$ , i.e.  $E(\mathcal{G}|_{\mathcal{C}}) = \bigcup_{x \in \mathcal{C}} E(\mathcal{G}_x^{\text{loc}})$ . Therefore, even if the total information  $M$  in Definition 6 cannot be accessed in practice,  $\mathcal{G}$  is still possible to be inferred via  $\mathcal{G}|_{\mathcal{C}}$ .

The other issue is that, in most cases, the output measurement is available rather than full-state measurements, which leads to the typical problem of network identifiability. A straightforward definition of identifiability of nonlinear Boolean dynamic network follows from Definition 5 and 6, where we verify linear network identifiability in a point-wise manner. Section III-A tells that the nonlinear Boolean dynamic network is defined on  $\{Q_x(q), P_x(q), H_x(q)\}_{x \in M}$ , whose model structures can be denoted by  $\{\mathcal{M}_x(\theta)\}_{x \in M}$ . The ground truth value of  $\theta$  may be different at each  $x \in M$ , which is denoted by  $\theta^*(x)$ .

**Definition 7.** Given the model structures  $\mathcal{M}_x(\theta)$  for each point  $x \in M$ , the nonlinear Boolean dynamic network is globally (or locally) identifiable at  $\theta^*(x)$  if, for any  $x^* \in M$ , each linear Boolean dynamic network with  $\mathcal{M}_x(\theta)$  is globally (or locally) identifiable at  $\theta^*(x^*)$ .

This definition helps us to take advantage of well-known results on network identifiability. For example, if we apply ARX models for  $x \in \mathcal{C}$ , and each  $x$  has a corresponding quadratic program, we know the nonlinear Boolean network is identifiable by definition, if the ground truth appears in the model set.

### IV. APPROXIMATED RECONSTRUCTION METHODS

According to Definition 6, the exact inference should be done by inferring  $(Q_{x_t}, P_{x_t}, H_{x_t})$  by using the input-output measurement in a “small enough” neighborhood of  $x_t$ , and then taking union of their corresponding Boolean networks. However, this is not possible in practice due to two major issues: 1) no enough data for locally defined dynamic networks at each value of state variables; 2) not robust in the presence of noise. With regard to the first issue, we divide the whole time series into several segments, each of which is used for an approximated inference of a Boolean network at a specific point  $x_{t_k}$ . The second issue is addressed by adopting a technique dealing with heterogeneity [6] to guarantee the robustness by inferring all local networks  $(Q_{x_t}, P_{x_t}, H_{x_t})$  together. Now this suggested treatment leads to an approximated inference procedure in general.

Consider a time series  $\{y(t_i) : i \in \mathbb{N}\}$  from equidistant sampling. Suppose that we divide it into  $K$  segments  $\{\{y(t_j)\}_{s_{k-1} \leq j \leq s_k} : k = 1, \dots, K; s_{k-1} \leq s_k \in \mathbb{N}\}$ , each of which corresponds to a neighborhood of  $x_{t_k}^*$  (of which we may not know the value) and may not have the same length. Let  $(Q_{x_{t_k}^*}, P_{x_{t_k}^*}, H_{x_{t_k}^*})$  denote the DSF locally defined for the segment  $(y(t_{s_{k-1}}), y(t_{s_k}))$ . For simplicity, we denote this

as  $(Q_k, P_k, H_k)$  instead. As presented in [6], in the sense of Prediction Error Minimization (PEM), the inference of linear dynamic network can be formulated as  $p$  independent pseudo-linear regression problems. Once we solve these  $p$  regression problems, we obtain the DSF  $(Q, P, H)$  and thus  $\mathcal{N}$ . See [6] for details on how the regressor and parameter variables are constructed from network model structures (i.e. the parametrization of DSF, e.g. ARX, ARMAX, etc.). In the following we will manifest how to integrate multiple linear network inference problems into one regression problem so as to improve robustness.

Without loss of generality<sup>3</sup>, we address the subproblem corresponding to the  $i$ -th output variable  $y_i$ . Let  $\mathbf{y}^{[k]}$  denote the vector of one-step predictions of  $y_i$  at time instants from  $t_{s_{k-1}}$  to  $t_{s_k}$ , and  $\mathbf{w}^{[k]}$  be the parameters of the approximated linear model at  $x_{t_k}^*$ . The pseudo-linear regression equation follows (see [6])

$$\hat{\mathbf{y}}^{[k]} = \mathbf{A}^{[k]}(\mathbf{w}^{[k]}) \mathbf{w}^{[k]}, \quad k = 1, \dots, K, \quad (11)$$

which can be rewritten into block matrices according to the ‘‘physical’’ meaning of each sub-vector of  $\mathbf{w}^{[k]}$

$$\begin{aligned} \mathbf{A}^{[k]} &\triangleq \begin{bmatrix} \mathbf{A}_{:,1}^{[k]} & \mathbf{A}_{:,2}^{[k]} & \dots & \mathbf{A}_{:,N}^{[k]} \end{bmatrix}, \\ \mathbf{w}^{[k]} &\triangleq \begin{bmatrix} (\mathbf{w}_1^{[k]})^T & (\mathbf{w}_2^{[k]})^T & \dots & (\mathbf{w}_N^{[k]})^T \end{bmatrix}^T, \end{aligned} \quad (12)$$

where  $p$  and  $m$  are the dimensions of output and input variables respectively, and  $N$  is the number of blocks. By optimizing  $\min_{\mathbf{w}^{[k]}} \|\mathbf{y}^{[k]} - \hat{\mathbf{y}}^{[k]}\|$  (or including penalty of network sparsity), we get an estimate of  $\mathbf{w}^{[k]}$  in the sense of PEM, which yields the linear dynamic network  $(Q_k, P_k, H_k)$  defined at  $x_{t_k}^*$ . It is not recommended to solve the same problem for each segment and then take union due to the algorithmic robustness in the presence of process noise.

Now consider improving the algorithmic robustness by inferring all local networks simultaneously to take advantage of the whole dataset. Let

$$\mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_k^{[1]} \\ \vdots \\ \mathbf{w}_k^{[K]} \end{bmatrix}, \quad \mathbf{w} \triangleq \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{bmatrix}$$

and integrate all segments by stacking (11) of each segment and rearranging blocks of matrices, yielding (13), and use  $\hat{\mathbf{y}} = \mathbf{A}(\mathbf{w})\mathbf{w}$  to denote (13b). Furthermore, we introduce two terms of group sparsity

$$\begin{aligned} \mathbf{w}^E &:= [\|\mathbf{w}_1^{[1]}\|_2, \dots, \|\mathbf{w}_1^{[K]}\|_2, \dots, \|\mathbf{w}_N^{[1]}\|_2, \dots, \|\mathbf{w}_N^{[K]}\|_2]^T, \\ \mathbf{w}^S &:= [\|\mathbf{w}_1\|_2, \dots, \|\mathbf{w}_N\|_2]^T, \end{aligned} \quad (14)$$

in which  $\mathbf{w}^E \in \mathbb{R}^{KN}$ ,  $\mathbf{w}^S \in \mathbb{R}^N$ ,  $\|\cdot\|_2$  is the  $l_2$ -norm of vectors. One may have noticed that  $\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2 + \lambda \|\mathbf{w}^E\|_0$  ( $\lambda \in \mathbb{R}^+$  is the regularization parameter) is almost equivalent to infer sparse local dynamic networks  $(Q_k, P_k, H_k)$  ( $k = 1, \dots, K$ ) separately. Furthermore, as studied in [6], the estimates via  $\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2 + \lambda \|\mathbf{w}^S\|_0$

<sup>3</sup>The whole network inference problem is solved by repeating the following operations on each element of the output variable  $y$ .

guarantees both network sparsity and consistent network topology for all datasets (i.e.  $K$  segments), where the latter means  $\mathcal{G}(Q_k, P_k, H_k) \equiv \mathcal{G}^0, \forall k$ , in which  $\mathcal{G}(Q_k, P_k, H_k)$  denotes the Boolean network determined from  $(Q_k, P_k, H_k)$  and  $\mathcal{G}^0$  denotes any fixed Boolean network. However, Definition 6 implies  $(Q_k, P_k, H_k)$  could have the different network topology. Thus, we combined these two penalty terms to allow both multiple network topology and algorithmic robustness

$$\underset{\mathbf{w}}{\text{minimize}} \|\mathbf{y} - \mathbf{A}(\mathbf{w})\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}^E\|_0 + \lambda_2 \|\mathbf{w}^S\|_0, \quad (15)$$

where  $\lambda_1, \lambda_2 \in \mathbb{R}^+$  are the regularization parameters. There is a trade-off between the two penalty terms:  $\lambda_1 \|\mathbf{w}^E\|_0$  promotes different sparse networks;  $\lambda_2 \|\mathbf{w}^S\|_0$  promotes  $(Q_k, P_k, H_k)$  ( $k = 1, \dots, K$ ) to have the same network topology. Assuming that the networks  $(Q_k, P_k, H_k), k = 1, \dots, K$  share most arcs in common, the algorithmic robustness can be understood in the following sense: due to  $\lambda_2 \|\mathbf{w}^S\|_0$ , the additional arcs of  $(Q_k, P_k, H_k)$  are included in the result only if it significantly contributes to decreasing the prediction error. This is also the reason why we emphasize that the presented approach is an approximated method. We increase the algorithmic robustness to noise by increasing the risk of missing arcs, which are not ‘‘significant’’ in the sense of predictability.

## V. ALGORITHMS

This section provides specific numerical algorithms to solve (15). We use the ARX parametrization as an example, in which  $\mathbf{A}$  does not depend on  $\mathbf{w}$  in (15). There are multiple way to heuristically solve convex-cardinality problems, e.g.  $l_1$ -norm heuristic approaches, *sparse Bayesian learning*. Here we give a solution using the classical  $l_1$ -heuristic treatment, which leads to the group LASSO problem [16]. To enhance sparsity further, one may use *Iterative Reweighted  $l_1$  Method* (see [17]), which is presented as follows

$$\begin{aligned} \mathbf{w}^{l+1} &= \arg \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \lambda_1 \sum_{i=1}^N \sum_{j=1}^K \mu_{i,j}^l \sqrt{\rho_{i,j}^E} \\ &\quad \|\mathbf{w}_i^{[j]}\|_2 + \lambda_2 \sum_{i=1}^N \nu_i^l \sqrt{\rho_i^S} \|\mathbf{w}_i\|_2, \end{aligned} \quad (16)$$

where  $\rho_{i,j}^E$  denotes the dimension of  $\mathbf{w}_i^{[j]}$ ,  $\rho_i^S$  denotes the dimension of  $\mathbf{w}_i$ ,  $l$  is the index of iterations,

$$\mu_{i,j}^l = \left[ \|\mathbf{w}_i^{[j]}\|_2 + \epsilon^l \right]^{-1}, \quad \nu_i^l = \left[ \|\mathbf{w}_i\|_2 + \epsilon^l \right]^{-1},$$

and  $\{\epsilon^l\}$  is a sequence converging to zero (see [17]), e.g.  $\epsilon^l \in (0, 1)$  is reduced by a factor of 10 until reaching the minimum of  $10^{-12}$ .

For large-dimensional problems of (16), in the  $l$ -th iteration, we present algorithms using *Proximal Operators* and *ADMM*. Let  $k$  denote the inner-loop iteration index for numerical optimization. Consider (16) as  $\min_{\mathbf{w}} f(\mathbf{w}) + g(\mathbf{w})$ , where  $f(\mathbf{w}) \triangleq (1/2) \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2$  and  $g(\mathbf{w})$  denotes the rest. Here we add  $1/2$  term in  $f(\mathbf{w})$  to simplify calculations, where the values of  $\lambda_1, \lambda_2$  are adjusted correspondingly. Given  $\nabla f(\mathbf{w}) = \mathbf{A}^T(\mathbf{A}\mathbf{w} - \mathbf{y})$ , the *Proximal Gradient Method* is to update  $\mathbf{w}$  by  $\mathbf{w}^{k+1} = \text{prox}_{\gamma g}(\mathbf{w}^k -$



$$\begin{bmatrix} \hat{\mathbf{y}}^{[1]} \\ \vdots \\ \hat{\mathbf{y}}^{[K]} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_{:,1}^{[1]} & \dots & \mathbf{A}_{:,N}^{[1]} & & & \\ & & & \ddots & & \\ & & & & \mathbf{A}_{:,1}^{[K]} & \dots & \mathbf{A}_{:,N}^{[K]} \end{bmatrix}}_{K \text{ Blocks}} \begin{bmatrix} \mathbf{w}^{[1]} \\ \vdots \\ \mathbf{w}^{[K]} \end{bmatrix} \quad (13a)$$

$$= \underbrace{\begin{bmatrix} \mathbf{A}_{:,1}^{[1]} & & & & & \\ & \ddots & & & & \\ & & \mathbf{A}_{:,1}^{[K]} & & & \\ & & & \dots & & \\ & & & & \mathbf{A}_{:,N}^{[1]} & \\ & & & & & \ddots \\ & & & & & & \mathbf{A}_{:,N}^{[K]} \end{bmatrix}}_{N \text{ Blocks}} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \end{bmatrix} \quad (13b)$$

$\gamma \nabla f(\mathbf{w}^k)$ ,  $\gamma \in \mathbb{R}_+$ , where  $k$  denotes the iteration index in proximal methods. Thus, the key step is to calculate the proximal operator  $\text{prox}_{\gamma g}(\mathbf{v})$ . It is easy to see that  $g(\mathbf{w}) = \sum_{i=1}^N g_i(\mathbf{w}_i)$ , where  $g_i(\mathbf{w}_i) \triangleq g_i^E(\mathbf{w}_i) + g_i^S(\mathbf{w}_i) := \lambda_1 \sum_{j=1}^K \mu_{i,j}^L \sqrt{\rho_{i,j}^E} \|\mathbf{w}_i^{[j]}\|_2 + \lambda_2 \nu_i^L \sqrt{\rho_i^S} \|\mathbf{w}_i\|_2$ . Therefore,  $\text{prox}_{\gamma g}(\mathbf{v}) = [\text{prox}_{\gamma g_1}(\mathbf{v}_1)^T \dots \text{prox}_{\gamma g_N}(\mathbf{v}_N)^T]^T$ , where  $\mathbf{v}$  is partitioned in the same way as  $\mathbf{w}$  in terms of  $\mathbf{w}_i, i = 1, \dots, N$ . However, it is difficult to calculate  $\text{prox}_{\gamma g_i}(\mathbf{v}_i)$  analytically. The *Dykstra-like Proximal Algorithm* in [18] presents a numerical way to calculate this proximal operator, as shown in Algorithm 1. The proximal

---

**Algorithm 1** Dykstra-like proximal algorithm for  $\text{prox}_{\gamma g_i}(\mathbf{v}_i)$ 


---

- 1: input:  $\mathbf{v}_i$
  - 2: output:  $\mathbf{w}_i = \text{prox}_{\gamma g_i}(\mathbf{v}_i)$
  - 3: parameters:  $\lambda_1, \lambda_2, \gamma, \rho_i^S, \rho_{i,j}^E, j = 1, \dots, K$
  - 4: **given** initial values:  $\mathbf{w}_i^0 = \mathbf{v}_i, \mathbf{r}^0 = 0, \mathbf{z}^0 = 0$
  - 5: **repeat**
  - 6:  $\mathbf{s}^k \leftarrow \text{prox}_{\gamma g_i^S}(\mathbf{w}_i^k + \mathbf{r}^k)$
  - 7:  $\mathbf{r}^{k+1} \leftarrow \mathbf{w}_i^k + \mathbf{r}^k - \mathbf{s}^k$
  - 8:  $\mathbf{w}_i^{k+1} \leftarrow \text{prox}_{\gamma g_i^E}(\mathbf{s}^k + \mathbf{z}^k)$
  - 9:  $\mathbf{z}^{k+1} \leftarrow \mathbf{s}^k + \mathbf{z}^k - \mathbf{w}_i^{k+1}$
  - 10: **until**  $\mathbf{w}_i$  converges
- 

operators in line 6 & 8 in Algorithm 1 can be calculated by *block soft thresholding* [19], shown as follows

$$\begin{aligned} \text{prox}_{\gamma g_i^S}(\mathbf{x}) &= \left(1 - \gamma \nu_i \lambda_2 \sqrt{\rho_i^S} / \|\mathbf{x}\|_2\right)_+ \mathbf{x}, \\ \text{prox}_{\gamma g_i^E}(\mathbf{x}) &= \left[\text{prox}_{\gamma g_{i,j}^E}(\mathbf{x}_1)^T \dots \text{prox}_{\gamma g_{i,j}^E}(\mathbf{x}_K)^T\right]^T, \\ \text{prox}_{\gamma g_{i,j}^E}(\mathbf{x}_j) &= \left(1 - \gamma \lambda_1 \mu_{i,j} \sqrt{\rho_{i,j}^E} / \|\mathbf{x}_j\|_2\right)_+ \mathbf{x}_j, \end{aligned} \quad (17)$$

where  $(\cdot)_+$  replaces each negative elements with 0, and  $\mathbf{x}$  is partitioned in the same way as  $\mathbf{w}_i$  in terms of  $\mathbf{w}_i^{[j]}, j = 1, \dots, K$ . Now we have got  $\text{prox}_{\gamma g}(\mathbf{w})$  to run (*Accelerated Proximal Gradient Method* (see [6] or [19])).

To implement ADMM, the proximal operator of  $f(\mathbf{w})$  needs to be calculated as

$$\text{prox}_{\gamma f}(\mathbf{v}) = (I + \gamma \mathbf{A}^T \mathbf{A})^{-1} (\gamma \mathbf{A}^T \mathbf{y} + \mathbf{v}). \quad (18)$$

Given the proximal operators of  $f(\mathbf{w})$  and  $g(\mathbf{w})$ , the ADMM algorithm is presented in Algorithm 2 (one may choose  $\gamma$  by line search methods for proximal gradient methods [19]).

---

**Algorithm 2** ADMM method

---

- 1: Precompute  $\mathbf{A}^T \mathbf{A}$  and  $\mathbf{A}^T \mathbf{y}$
  - 2: **given** an initial value  $\mathbf{w}^0, \mathbf{v}^0, \mathbf{u}^0, \gamma^0 = 1, \beta = 1/2$
  - 3: **repeat**
  - 4: Let  $\gamma \leftarrow \gamma^k$
  - 5: **repeat**
  - 6:  $\hat{\mathbf{w}} \leftarrow \text{prox}_{\gamma f}(\mathbf{v}^k - \mathbf{u}^k)$  using (18)
  - 7: **break** if  $f(\hat{\mathbf{w}}) \leq f(\mathbf{w}^k) + \nabla f(\mathbf{w}^k)^T (\hat{\mathbf{w}} - \mathbf{w}^k) + (1/2\gamma) \|\hat{\mathbf{w}} - \mathbf{w}^k\|_2^2$
  - 8:  $\gamma \leftarrow \beta \gamma$
  - 9: **until** ;
  - 10:  $\mathbf{w}^{k+1} \leftarrow \hat{\mathbf{w}}, \gamma^{k+1} \leftarrow \gamma$
  - 11: \*Compute  $\text{prox}_{\gamma g_i}(\mathbf{w}_i^{k+1} + \mathbf{u}_i^k)$  using Algorithm 1 for all  $i = 1, \dots, N$
  - 12:  $\mathbf{v}^{k+1} \leftarrow \text{prox}_{\gamma g}(\mathbf{w}^{k+1} + \mathbf{u}^k)$
  - 13:  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k + \mathbf{w}^{k+1} - \mathbf{v}^{k+1}$
  - 14: **until** any standard stopping criteria
    - ▷ \*Step 11 should be implemented in parallel if possible.
- 

## VI. NUMERICAL EXAMPLES

This section uses the Millar-10 model as an example to perform the proposed method. The complete nonlinear ordinary differential equations have 19 state variables (mRNAs and proteins), of which 8 mRNAs and 1 protein are output variables (LHY, TOC1, Y, PPR9, PPR7, NI, GI, ZTL), which can be measured in biological experiments (see the supplement of [7] for details). The model simulates the behaviors of eukaryotic circadian clocks, whose outputs are sampled by 30 min, plotted in Figure 2. The light is controlled as follows: -48h~0h is two cycles of light-dark, each of which lasts for 12 hours; 0h~48h is always in light, illustrated as a square wave in Figure 2. The ground truth of interaction path diagram of key mRNAs for circadian clocks is illustrated in Figure 3a.

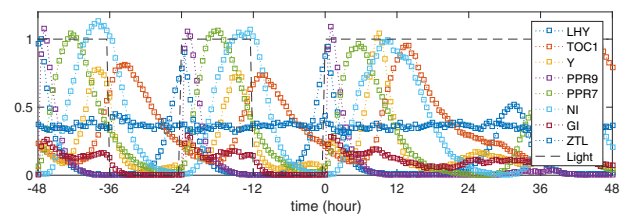


Fig. 2: This simulation uses the Millar-10 model to simulate the measurements of key mRNAs for eukaryotic circadian clocks in experiments.

The proposed method is used to perform network reconstruction. The whole time series is divided equidistantly into 1 (i.e. applying linear approach), 2, 3, 4 segments, i.e.  $K = 1, 2, 3, 4$ . The results of  $K = 1, 2, 3$  is shown in Figure 3. To determine whether an edge exists or not, we check the 2-norm of all parameters in its associated transfer function. Note that our approach has enhanced sparsity, and thus it is no longer an issue as other methods to select the threshold of 2-norm<sup>4</sup> to determine whether an edge should exist or not, e.g. most methods reviewed in [20]. Following the conventions on comparing different algorithms, the performance curves of the algorithm are given in Figure 4.

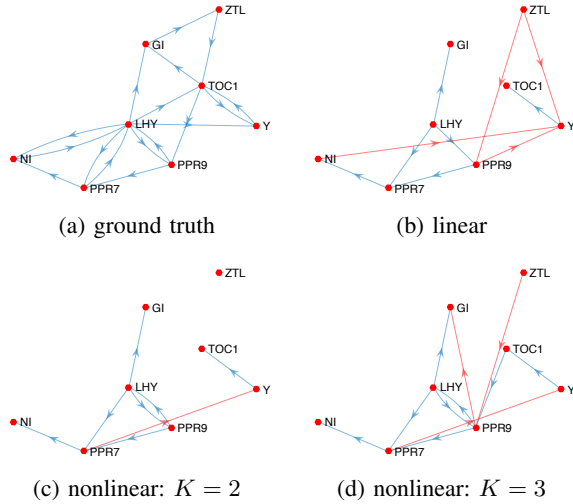


Fig. 3: Examples of network inference results.

One problem deserving to be explained is why there are still several links missed. These links are introduced by the nonlinear terms combining  $x$  and  $u$ , i.e. nonlinear terms as  $f(\dots, x_k, \dots; u)$ , which has not been included in our extension. The bottom of this issue is how to extend the DSF to include the bilinear terms of  $x$  and  $u$ .

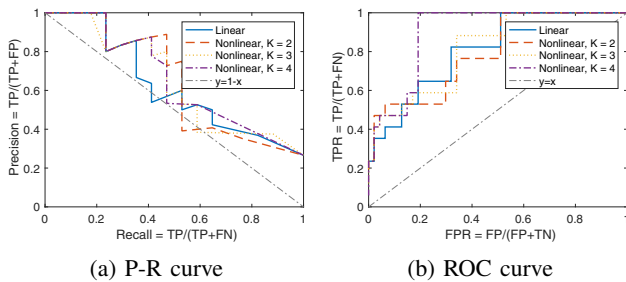


Fig. 4: Performance curves: the precision-recall and the ROC curve. They are obtained by setting different model orders (ranging from 1 to 48), counting how many times each edge appears in all inference results and then choosing different thresholds of frequencies of edges appearing in total.

<sup>4</sup>Here we use  $1e-6$ . The threshold  $1e-8$  or even  $1e-10$  gives almost the same results.

## VII. CONCLUSIONS

The paper provides a way to define Boolean dynamic networks for a certain class of nonlinear dynamical systems. Based on this definition, an inference method was introduced with great practical applicability. To handle large-scale problems, an ADMM algorithm is presented. The Millar-10 model in systems biology is used as an numerical example in network reconstruction. Besides the Boolean dynamic networks for nonlinear systems, we also provide a definition of linear network identifiability to unify different perspectives on network identifiability used in available literature.

## REFERENCES

- [1] C. Hsiao, "Autoregressive modeling and causal ordering of economic variables," *Journal of Economic Dynamics and Control*, vol. 4, pp. 243–259, 1982.
- [2] M. Eichler, "Granger causality and path diagrams for multivariate time series," *Journal of Econometrics*, vol. 137, pp. 334–353, 2007.
- [3] J. Goncalves and S. Warnick, "Necessary and Sufficient Conditions for Dynamical Structure Reconstruction of LTI Networks," *Automatic Control, IEEE Transactions on*, vol. 53, no. 7, pp. 1670–1674, 2008.
- [4] H. H. M. Weerts, A. G. Dankers, and P. M. J. Van den Hof, "Identifiability in dynamic network identification," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1409–1414, 2015.
- [5] A. Chiuso and G. Pillonetto, "A Bayesian approach to sparse dynamic network identification," *Automatica*, vol. 48, no. 8, pp. 1553–1565, 2012.
- [6] Z. Yue, W. Pan, J. Thunberg, L. Ljung, and J. Goncalves, "Linear Dynamic Network Reconstruction from Heterogeneous Datasets," in *Preprints of the 20th World Congress, IFAC*, Toulouse, France, 2017, pp. 11 075–11 080.
- [7] A. Pokhilko, S. K. Hodge, K. Stratford, K. Knox, K. D. Edwards, A. W. Thomson, T. Mizuno, and A. J. Millar, "Data assimilation constrains new connections and components in a complex, eukaryotic circadian clock model," *Molecular systems biology*, vol. 6, no. 1, p. 416, 2010.
- [8] W. Pan, Y. Yuan, J. Goncalves, and G.-B. Stan, "A Sparse Bayesian Approach to the Identification of Nonlinear State-Space Systems," *Automatic Control, IEEE Transactions on*, vol. 61, no. 1, pp. 182–187, 2016.
- [9] W. Pan, Y. Yuan, L. Ljung, J. Gon, and G.-B. Stan, "Identifying biochemical reaction networks from heterogeneous datasets," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2525–2530.
- [10] J.-F. L. Gall, *Brownian Motion, Martingales, and Stochastic Calculus (Graduate Texts in Mathematics)*, 1st ed. Springer, 2016.
- [11] D. Hayden, Y. Yuan, and J. Goncalves, "Network Identifiability from Intrinsic Noise," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, p. 1, 2016.
- [12] L. Ljung, *System Identification: Theory for the User*, ser. Prentice-Hall information and system sciences series. Prentice Hall PTR, 1999.
- [13] Z. Yue, J. Thunberg, L. Ljung, and J. Goncalves, "On Definition and Inference of Nonlinear Boolean Dynamic Networks," *arXiv*, 2017.
- [14] M. Gevers, A. S. Bazanella, and A. Parraga, "Structural conditions for the identifiability of dynamical networks," in *IEEE Conference on Decision and Control (preprint)*, 2016.
- [15] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed., ser. Graduate Texts in Mathematics 218. Springer-Verlag New York, 2012.
- [16] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [17] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.
- [18] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.
- [19] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [20] A. Aderhold, D. Husmeier, and M. Grzegorzczak, "Statistical inference of regulatory networks for circadian regulation," p. 227, 2014.