Data-Driven Network Models for Genetic Circuits From Time-Series Data with Incomplete Measurements

Enoch Yeung[1], Jongmin Kim[2], Ye Yuan[3], Jorge Gonçalves[4], and Richard M. Murray[5,6]

***Abstract*—Synthetic gene networks are frequently conceptualized and visualized as static graphs. This view of biological programming stands in stark contrast to the transient nature of biomolecular interaction, which is frequently enacted by labile molecules that are often unmeasured. Thus, the network topology and dynamics of synthetic gene networks can be difficult to verify *in vivo* or *in vitro*, due to the presence of unmeasured biological states. Here we introduce the dynamical structure function as a new mesoscopic, data-driven class of models to describe gene networks with incomplete measurements. We introduce a network reconstruction algorithm and a code base for reconstructing the dynamical structure function from data, to enable discovery and visualization of graphical relationships in a genetic circuit diagram as *time-dependent functions* rather than static, unknown weights. We prove a theorem, showing that dynamical structure functions can provide a data-driven estimate of the size of crosstalk fluctuations from an idealized model. We illustrate this idea with numerical examples. Finally, we show how data-driven estimation of dynamical structure functions can explain failure modes in two experimentally implemented genetic circuits, a historical genetic circuit and a new *E. coli* based transcriptional event detector.**

I. Introduction

Synthetic gene networks fulfill diverse roles in realizing circuit logic [1] and timing in living organisms [2]. Ranging from single-input inverters [3], [4] to combinatorial input logic gates [5], [6], reduction in DNA synthesis and sequencing costs have made it possible to build increasingly complex genetic circuits with tens to hundreds of components. However, the ability to build novel biological circuitry often outpaces our ability to revise designs or to take verify what has been built behaves as intended. As the fields of synthetic and systems biology continue to build and integrate on successes of circuit and device-level complexity to engineer entire genetic systems or pathways, we are consistently seeing failure modes that arise from a lack of modularity, retroactivity [7], [8], [9], and context effects [10].

Likewise, the expansion of CRISPR-based methods for genome editing and targeted gene knockdown [11], [12] has enabled a broader category of systems biology design problems, centered on redesigning genomes [13] or reprogramming host regulatory networks [14] to target specific environmental niches or to exhibit a particular phenotype. The underlying genetic program implicit in these systems biology objectives is often a vast, complex, and dynamic network of interacting genes, mRNA, and proteins. The expansion in DNA sequencing read depth has made it possible to profile individual genes via the transcriptome [15], which combined with quantitative proteomics [16] or metabolomics [17], enables systems-level analysis of network activity. But prohibitive sampling and library preparation costs make obtaining highly time-resolved omics' measurements hard. This makes it difficult to infer dynamic network activity at the scale of whole cell models [18] without extensive experimental investment.

Dynamic network models that describe the intricate interactions between every biomolecular state or species are referred to as state-space models. Two key variables that often determine the behavior of these network models are its network topology [19], [20] and parametric realization [21], [22]. The structure of a network is generally determined by how states in the system causally affect each other [22]; edges in the network are determined by causal dependence while nodes are determined by the states of the system [23].

Identifying the active, dynamic network structure of a biological network is critical, since the hypothesized network architecture of a genetic circuit may be very different from the realized network architecture using a specific collection of parts, sequences, and composition approach [24]. While network structure alone does not determine dynamical behavior, though, parametric information is also important in determining what dynamical behaviors a system can achieve [25]. Rather, network structure, or topology, often defines or narrows the possible behaviors a system can achieve. Without any structural constraints, a dynamical system can have arbitrary input-output behavior. Once network structure is imposed, the set of realizable input-output trajectories can be reduced [26], [27]. If the realized network differs significantly from the intended network design, the dynamics of the system may produce faults or glitches when appropriately excited or interrogated [28], [29]. Getting the actual network topology to match the intended network motif is thus a key element to robust synthetic biological design.

In systems and synthetic biology, canonical network motifs are broadly accepted as enabling useful dynamical behavior [27], [30]. For example, an incoherent feedforward loop can

1-Center for Biological Engineering, Biomolecular Science and Engineering Program, Department of Mechanical Engineering, Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara , 2- Department of Life Sciences, POSTECH 3- School of Artificial Intelligence & Automation, Hua Zhong University of Science and Technology 4-Systems Biology Research Group, University of Luxembourg, 5-Control and Dynamical Systems, California Institute of Technology, 6-Biology and Biological Engineering, California Institute of Technology. Please address all correspondence to eyeung@ucsb.edu

be used for fold-change detection or adaptation [31], [32]. A cyclic network of repressors is associated with either oscillations [33], [34], [35] or multi-stability [36] while a dual negative feedback network of two nodes is used as memory module or toggle switch [37]. Still, the active, dynamic network architecture of most realizations of these network motifs in the form of genetic circuits are not formally characterized or catalogued [38]. Systematic, generalizable tools that can discover and model dynamic network topology from data are valuable. [1].

Circuit network discovery is, at its core, a network reconstruction problem. Given a desired network motif and a physical system, we need to use measurements of the system to determine if the actual, active network of the system matches the intended design. There have been many network reconstruction algorithms developed for natural and synthetic biological networks [39], [40], [24], [41], [42], [43], [44], [45]. Historically, the approach to discovering network interactions has involved direct perturbation of biochemical species or components in a network [41], [45], [46]. Individual nodes are perturbed and depending on if nearby nodes positively or negatively correlate, an activating or repressing relationship between two network nodes can be inferred. In [24], this framework was taken a step further, by showing that the behavior of direct and indirect links in a benchmark circuit is network topology dependent. This provided a means for using steady state perturbation data [39], [47], [45] to estimate network models. Further, these steady-state estimation algorithms have been verified using a benchmark synthetic gene cirucit [43]. More recently, the authors in [42] and [40] showed that retroactivity in gene networks can paradoxically confound network predictions that are based wholly on correlation measures. The core issue is that even when measurement data for all biological states is available, causality is difficult to determine from steady-state measurement data affected by back-action or retroactivity in genetic networks [40].

At the single cell level, the reconstruction problem for biological networks introduces challenges of inferring nonlinear stochastic models from noisy data [48], [49], [44]. In [48], the authors show that by comparing average abundances, molecule lifetimes, covariances, and magnitude of step, they can map pairwise interaction dynamics, even when the rest of the system is completely unspecified. The key observation is that assembly stoichiometry of new molecules is fixed, so unbalanced production of linked precursor components will exacerbate imbalance further, resulting in empirically observed large fluctuations. Further, Hilfinger, Norman and Paulsson show in [49] that there are statistical invariants for certain kinds of network interactions, which can be used to evaluate and challenge existing hypotheses of stochastic gene interaction. More recently, Wang, Lin, Sontag and Sorger show that effective stoichiometric spaces can be used to determine network structure from the covariances of single-cell multiplex data [44]. These studies show that it is possible to infer meaningful structural information about a genetic network, even only when a portion of the network states are observed and the data is fundamentally noisy.

In this paper we introduce a class of mesoscopic network reconstruction models with adaptable resolution, commensurate with the depth or coverage of the circuit states (or genome) available from fluorimetric, spectometry-based, or sequencing based measurements. Our method is distinct in that we consider the use of high-resolution time-series data, but where only partial measurement of the network's nodes is feasible. Further, we consider dynamic measurements of bulk culture rather than single cell, where we benefit from the assumptions of high molecular copy number and large reaction volumes [49]. Specifically, we present the dynamical structure function, an abstract model class from linear time-invariant systems theory and show it can be used as a generalized representation of measured interactions between biological or biochemical states. The contributions of this paper are: 1) we show how a dynamical structure function can encode both direct and crosstalk network interactions, by way of theorem and simulated examples, 2) we develop a direct estimation algorithm and code to directly estimate the dynamical structure function, as well as visualization tools to monitor repression and activation in genetic circuits, and 3) we demonstrate this theory on two experimental systems: A) an *in vitro* genelet repressilator from the synthetic biology literature and B) a novel transcriptional event detector that we build specifically to illustrate dynamical structure reconstruction.

II. Representing Network Interactions in Partially Measured Biological Networks with Dynamical Structure Functions

The tradeoffs between cost of network reconstruction and the "informativity" of the structural representation are especially clear in synthetic and systems biology research. In this area, finding or verifying the network of a biological system is an important problem. However, discovering the entire chemical reaction network is typically an ill-posed problem, since additional reactions may be introduced due to host or environmental context [52], loading effects, or unanticipated retroactivity effects [53], [54], [55], [56], [7]. Even without these effects, the reconstruction problem is equivalent to finding a unique realization for the dynamical system from direct measurements of every state of the system. Unique realization problems are difficult, unless the system of interest has specific structure, e.g., measurement functions of the state that are diffeomorphic [57], [58]. On the other hand, there are many inputs that can be used to perturb the system of interest, e.g. silencing RNA [59], genetic knock-outs [60], and small chemical inducers [61]. Using these inputs, it is straightforward to reconstruct the transfer function of the system [73], [51]. However, the transfer function contains virtually no information about how chemical species within the system are interacting.

An intermediate representation of network structure that addresses this trade-off is the dynamical structure function [62]. The dynamical structure function is a representation derived from linear systems theory, and thus can be used to model transients of a genetic circuit around an operating point or even unstable network dynamics diverging from an equilibrium point. It is a more detailed description of network

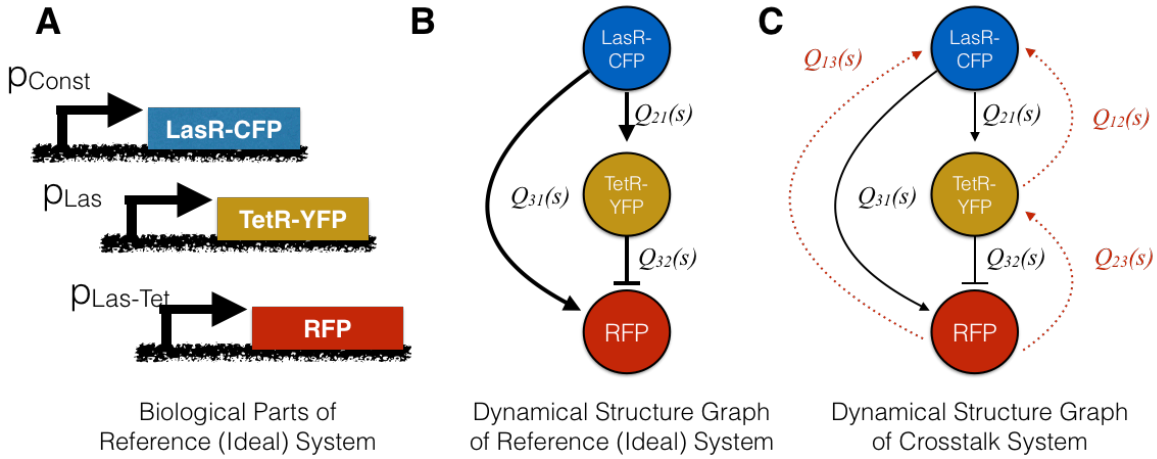

Fig. 1. **Dynamical structure functions can be used to analyze synthetic gene networks: (A)** Synthetic biological parts for a incoherent feedforward loop (IFFL) using the LasR activator, the TetR repressor, and reporter proteins CFP, YFP, and RFP. **(B-C)** The dynamical structure graphs of the crosstalk-free IFFL from system (6), in **(B)** and the crosstalk-impacted IFFL from system (7), in **(C)**. Nodes represent measured biochemical species, with black edges denoting causal dependencies stemming from designed interactions, and red edges denoting causal dependencies arising from crosstalk or loading effects. Notice that the dynamical structure captures network models interactions that are not described by the system transfer function $G(s)$.

structure than the transfer function since it models the causal interactions between measured outputs, in addition to the causal dependencies of outputs on input variables. At the same time, it does not require complete state feedback for reconstruction, since it only models the interactions among output states. In biological systems, this is especially applicable since the output variables of a system are also a subset of the state variables. All unmeasured states are subsumed in the edge-weight functions that describe interactions between measured variables. It is thus possible to experimentally target specific biochemical species to measure and verify that the network structure of a biological system is functioning as intended. Most notably, necessary and sufficient conditions for recovery of dynamical network models have been developed and well-studied [62], [22], [63], [64], [65], [66], but so far no open-source algorithms, code bases, or applications of this theory have been developed directly for synthetic biology.

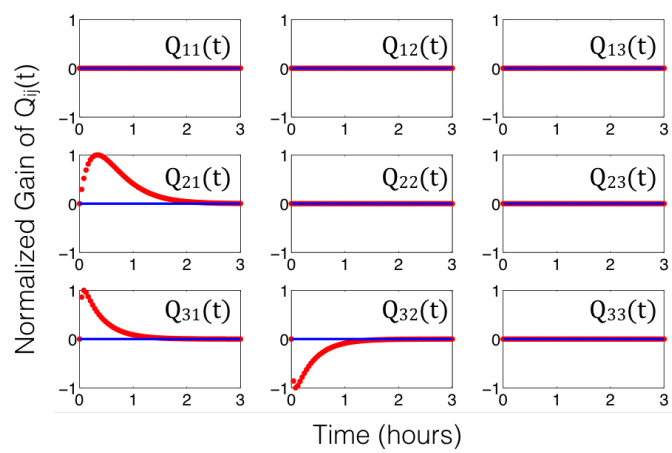

Fig. 2. **Dynamical structure functions describe how network structure evolves over time (and as a function of frequency):** The time-lapse response of the dynamical structure convolution kernel $Q^a(t) = \mathcal{L}^{-1}(Q^a(s))$ for the incoherent feedforward loop in system (6). By examining the functional response of each entry in $Q^a(t)$ (or $Q^a(s)$), we see that the network structure of the incoherent feedforward loop in Example II-A1 is a time-evolving, or *dynamic*, entity.

A. Dynamical Structure Functions

We briefly review the theory of dynamical structure functions, as they pertain to biochemical reaction networks. In practice, the state of the dynamical system $x = \begin{bmatrix} y^T & x_h^T \end{bmatrix}^T \in \mathbb{R}^n$, where $y \in \mathbb{R}^p$ are the measured chemical states of the dynamical system, corresponding to components of the biochemical reaction network tagged with fluorescent reporters, and $x_h \in \mathbb{R}^{n-p}$ are the unmeasured chemical states. It is also the cases that there are exogenous inputs $u \in \mathbb{R}^m$ that can be introduced to influence the dynamics of the state x. With the exception of oscillators, many biochemical reaction networks converge to a steady state. Moreover, it is generally the case that the parameters of biochemical reaction networks are time-invariant, so long as macroscopic experimental settings of the system such as temperature, growth media, and dissolved oxygen content remain fixed. Therefore, while the model of a biochemical reaction network is of the form

$$\begin{aligned} \dot{y} &= f_y(y, x_h, u), \quad y(0) = y_0 \\ \dot{x}_h &= f_{x_h}(y, x_h, u), \quad x_h(0) = x_{h,0}, \\ y &= \begin{bmatrix} \mathbf{I}_{p\times p} & 0 \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix} \end{aligned} \tag{1}$$

we will suppose that we can linearize the system about either an equilibrium point, a nominal operating point, or even an (unstable or oscillatory) initial condition to extract network dynamics. In biological systems, networks are almost never precisely linear, but we presume to model local fluctuations or perturbations from a target point in the state space. As we will see in the sequel, this will be enough to extract relevant network information. Proceeding with the linearization, we can

write the system in the form:

$$\begin{bmatrix} \dot{y} \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \qquad y = C \begin{bmatrix} y \\ x_h \end{bmatrix}. \tag{2}$$

where

$$C = \begin{bmatrix} \mathbf{I}_{p\times p} & 0 \end{bmatrix}.$$

We also assume the system's initial condition of the linearized system is $x(0) = 0$, and the entries in $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are calculated as

$$A_{11} \equiv \frac{\partial f_y(y,x_h,u)}{\partial y}|_{x=x_e,u=u_e}, \quad A_{12} \equiv \frac{\partial f_y(y,x_h,u)}{\partial x_h}|_{x=x_e,u=u_e}$$
$$A_{21} \equiv \frac{\partial f_{x_h}(y,x_h,u)}{\partial y}|_{x=x_e,u=u_e}, \quad A_{22} \equiv \frac{\partial f_{x_h}(y,x_h,u)}{\partial x_h}|_{x=x_e,u=u_e}$$
$$B_1 \equiv \frac{\partial f_y(y,x_h,u)}{\partial u}|_{x=x_e,u=u_e}, \quad B_2 \equiv \frac{\partial f_{x_h}(y,x_h,u)}{\partial u}|_{x=x_e,u=u_e}$$

Taking Laplace transforms, solving for $X_h(s)$ and replacing it in $Y(s)$ we obtain

$$sY = W(s)Y(s) + V(s)U(s) \tag{3}$$

where

$$W(s) = A_{11} + A_{12}(sI - A_{22})^{-1}A_{21} \qquad V(s) = B_1 + A_{12}(sI - A_{22})^{-1}B_2. \tag{4}$$

Defining $D(s) = \text{diag}\,(W(s))$ and subtracting $D(s)Y(s)$ from both sides of equation (3) and solving for $Y(s)$ we obtain the following equation

$$Y = Q(s)Y(s) + P(s)U(s) \tag{5}$$

where $Q(s) = (sI - D)^{-1}(W - D)$ is a $p \times p$ transfer function matrix and $P(s) = (sI - D)^{-1}V$ is a $p \times m$ transfer function matrix. Each entry $Q_{ij}(s)$ is a transfer function that describes the causal dependency of measured state $Y_i(s)$ on measured state $Y_j(s)$. Similarly, the transfer function $P_{ij}(s)$ describes the causal dependency of measured state $Y_i(s)$ on input $U_j(s)$. The matrix pair $(Q(s), P(s))$ is known as the *dynamical structure function*, where $Q(s)$ is referred to as the network structure and $P(s)$ as the control structure.

Note that $Q(s)$ is defined as $Q(s) = (sI - D)^{-1}(W - D)$ rather than $Q(s) = 1/sW(s)$. This guarantees that the diagonal entries of $Q(s)$ are 0, which implies that any non-zero terms $Q_{ij}(s)$ are A) strictly proper transfer functions and thus causal, B) descriptions of causal interactions among measured Y_i and Y_j. This also means that $Q(s)$, defined in this way, is unique and has p less parameters to identify on its diagonal. This construction of $Q(s)$ and $P(s)$ ultimately ensures identifiability [62] under reasonable assumptions of independent input perturbation [39], [40], [24], [41], [42], [43], [44], [45]. Further, if $Q(s) = W/s$, then we would face two simultaneous challenges in estimation 1) disentangling autoregulatory dynamics (Y_i to Y_i) from pairwise interactions (Y_i to Y_j), 2) too many unknown parameters in both $Q(s)$ and $P(s)$. Lastly, we find that studying the pairwise interactions $Q_{Ij}(s)$ can already elicit important functional information about a genetic network, as illustrated by the next two examples.

1) Example: The DSF of an Idealized Incoherent Feed-forward Loop: Consider the following synthetic biology design problem: design and implement an incoherent feed-forward loop. Specifically, we consider implementing a feed-forward loop using the synthetic parts pLac-LasR-CFP-LVA, pLas-TetR-YFP-LVA, and pLas-Tet-RFP-LVA and IPTG, $C_3O_6H_{12} - $HSL, and aTc as inputs (see Figure 1). We model the protein concentration of LasR-CFP, TetR-YFP, and RFP as x_1, x_2, and x_3, respectively. We denote the corresponding mRNA species for each of these proteins as m_1, m_2, and m_3. A simple model without any loading effects, describing the dynamics of these states can be written as:

$$\begin{aligned}
\dot{x}_1 &= \rho_1 m_1 - \frac{C_0 x_1/k_{1,d}}{1 + x_1/k_{1,d}} \\
\dot{x}_2 &= \rho_2 m_2 - \frac{C_0 x_2/k_{2,d}}{1 + x_2/k_{2,d}} \\
\dot{x}_3 &= \rho_3 m_3 - \frac{C_0 x_3/k_{3,d}}{1 + x_3/k_{3,d}} \\
\dot{m}_1 &= \frac{\alpha_1 u_1}{k_{M,u1} + u_1} - \delta_m m_1 \\
\dot{m}_2 &= \frac{\alpha_2 x_1 u_2/k_{M,u2}}{1 + x_1 u_2/k_{M,u2}} - \delta_m m_2 \\
\dot{m}_3 &= \frac{\alpha_3 x_1 u_2/k_{M,u2}}{1 + x_1 u_2/k_{M,u2} + x_2/\left(k_{M,2} + u_3/k_{M,u3}\right)} - \delta_m m_3 \\
y &= \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \vec{x}^T & \vec{m}^T \end{bmatrix}^T
\end{aligned} \tag{6}$$

The dynamical structure function for this system is derived by taking Laplace transforms and eliminating the hidden mRNA states of x_1, x_2, and x_3, namely m_1, m_2, m_3, see [62] or [66] for a detailed derivation of dynamical structure functions. The network and control structure matrix transfer functions are written $(Q^a(s), P^a(s))$ where $Q^a(s)$ is written as

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{0.045}{s^2+1.5\,s+5.7\cdot 10^{-4}} & 0 & 0 \\ \frac{1.5\cdot 10^{-4}}{s^2+1.7\,s+0.2} & -\frac{1.5\cdot 10^{-4}}{s^2+1.7\,s+0.2} & 0 \end{pmatrix}$$

and $P^a(s)$ is

$$\begin{pmatrix} \frac{6.7\cdot 10^{-7}}{s^2+1.5\,s+5.7\cdot 10^{-4}} & 0 & 0 \\ 0 & \frac{244.0}{s^2+1.5\,s+5.7\cdot 10^{-4}} & 0 \\ 0 & \frac{0.78}{s^2+1.7\,s+0.2} & \frac{7.8}{s^2+1.7\,s+0.2} \end{pmatrix}$$

The network, with edge weight functions corresponding to the entries of $Q^a(s)$, is drawn in Figure 1B. Notice that if we take $s \in \mathbb{R}_{>0}$, the sign of the entries in $Q^a(s)$ coincides with the form of transcriptional regulation implemented by TetR and LasR, respectively. In [67] it was shown that the sign definite properties of entries in $Q(\mathbb{R}_{>0})$ are useful for reasoning about the monotonicity of interactions between measured outputs and how fundamental limits in system performance relate to network structure.

Let us now consider the inverse Laplace transform of $\mathcal{L}^{-1}\left(Q^a(s)\right)$, we remark that $Y(t) = \int_0^t Q^a(t)Y(t-\tau)$ follows from the equation

$$\mathcal{L}^{-1}\left(Y(s)\right) = \mathcal{L}^{-1}\left(Q^a Y(s) + P^a U(s)\right)$$

whenever $u(t) \equiv 0$ such that $U(s)$ is 0. This argument holds in general for any system of the form (2). In particular, the

entries $Q^a(t)$ act as convolution kernels, and taken with the integral, define an operator for mapping $y_j(t)$ to $y_i(t)$. Most interestingly, we can see that the network structure of this incoherent feedforward loop is *dynamical*, hence our usage of the term *dynamical structure* function to describe the network structure among the measured chemical species $y(t)$. In this particular case, the time-domain analogue of the dynamical structure (or dynamical structure convolution kernel) is given as

$$Q^a(t) \equiv \begin{pmatrix} 0 & 0 & 0 \\ Q_{21}(t) & 0 & 0 \\ Q_{31}(t) & Q_{32}(t) & 0 \end{pmatrix}$$

where

$$Q_{21}(t) = \left(5.5 \cdot 10^{-3}\right) \mathrm{e}^{-\left(8.3 \cdot 10^{-4}\right) t} \sinh\left(\left(6.0 \cdot 10^{-5}\right) t\right)$$

$$Q_{31}(t) = \left(4.1 \cdot 10^{-7}\right) \mathrm{e}^{-\left(4.9 \cdot 10^{-3}\right) t} \sinh\left(\left(4.1 \cdot 10^{-3}\right) t\right)$$

$$Q_{32}(t) = -\left(9.7 \cdot 10^{-8}\right) \mathrm{e}^{-\left(4.9 \cdot 10^{-3}\right) t} \sinh\left(\left(4.1 \cdot 10^{-3}\right) t\right)$$

A visualization of each of these impulse kernel functions and their corresponding location in the dynamic adjacency matrix, defined by $Q^a(t)$, is given in Figure 2. Notice how the activating or repressing nature of genetic regulation is encoded by the positive or negative sign of the corresponding kernel response. In addition to uncovering the Boolean network of interactions between biological states, the dynamical network convolution kernel $Q^a(t)$ reveals the time-scales of response of each network edge, as well as the amplitude and the rate of decay of the gain from the time of impulse. Similar response profiles can be generated for step function inputs, though finite impulse inputs are typically more common in biological networks. Interestingly, the transfer function $G^a(s)$ of the system is likewise lower triangular, reflecting the feedforward network topology in the genetic circuit. Specifically, $G^a(s)$ has a sparsity structure of the form

$$G^a(s) \equiv \begin{bmatrix} G^a_{11}(s) & 0 & 0 \\ G^a_{21}(s) & G^a_{22}(s) & 0 \\ G^a_{31}(s) & G^a_{32}(s) & G^a_{33}(s) \end{bmatrix}.$$

2) Example: The DSF of an Incoherent Feedforward Loop with crosstalk: In prototyping a feedforward loop, it is important to anticipate *in vivo* context effects. We consider the same biocircuit as described in Example II-A1, except now we specifically consider loading effects frequently neglected in the design process of synthetic biology. First, we note that each gene may be susceptible to loading effects [7]. For each gene in Figure 1A, a degradation tag is added, to provide tunability, to the rate of degradation of the protein. Inside the cell, a protease called ClpXP targets these degradation tags and degrades the associated protein. Different tags can be incorporated to modulate the gain of the degradation process. Further, these degradation tags can be subject to mutagenesis experiments, as a means to modulate tunability.

Tunability of degradation introduces a tradeoff in performance. Since the ClpXP protease is a housekeeping protein expressed to form a common pool of proteases for all genes in the cell, there is a limit to the supply of free ClpXP protein in any instant of the cell's growth cycle. When there are too many degradation-tagged proteins [68], the overloading of the protein degradation queue can trigger unwanted effects such as stress response. More directly, the competition for scarce proteases can induce coupled dynamics or a *virtual* or *indirect* interaction between two genes competing for the same protease pool. Even if the genes were engineered to have no direct transcriptional or translational cross-regulation, the competition for the same protease effectively couples the protein states of both genes. Modifying the above model to account for these type of loading effects yields:

$$\begin{aligned}
\dot{x}_1 &= \rho_1 m_1 - \frac{C_0 x_1/k_{1,d}}{1 + x_1/k_{1,d} + x_2/k_{2,d} + x_3/k_{3,d}} \\
\dot{x}_2 &= \rho_2 m_2 - \frac{C_0 x_2/k_{2,d}}{1 + x_1/k_{1,d} + x_2/k_{2,d} + x_3/k_{3,d}} \\
\dot{x}_3 &= \rho_3 m_3 - \frac{C_0 x_3/k_{3,d}}{1 + x_1/k_{1,d} + x_2/k_{2,d} + x_3/k_{3,d}} \\
\dot{m}_1 &= \frac{\alpha_1 u_1}{k_{M,u1} + u_1} - \delta_m m_1 \\
\dot{m}_2 &= \frac{\alpha_2 (x_1 u_2/k_{M,u2})}{1 + x_1 u_2/k_{M,u2}} - \delta_m m_2 \\
\dot{m}_3 &= \frac{\alpha_3 x_1 u_2/k_{M,u2}}{1 + x_1 u_2/k_{M,u2} + x_2/\left(k_{M,2} + u_3/k_{M,u3}\right)} - \delta_m m_3 \\
y &= \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \vec{x}^T & \vec{m}^T \end{bmatrix}^T
\end{aligned} \tag{7}$$

Computing the dynamical structure function, we obtain $Q^c(s)$

$$\begin{pmatrix} 0 & \frac{1.6 \cdot 10^{-3}}{s + 2.1 \cdot 10^{-3}} & \frac{0.041}{s + 2.1 \cdot 10^{-3}} \\ \frac{\left(1.6 \cdot 10^{-3}\right) s + 0.048}{s^2 + 1.5\, s + 3.3 \cdot 10^{-3}} & 0 & \frac{0.041}{s + 2.1 \cdot 10^{-3}} \\ \frac{\left(3.8 \cdot 10^{-4}\right) s + 7.4 \cdot 10^{-4}}{s^2 + 1.6\, s + 0.13} & \frac{\left(3.8 \cdot 10^{-4}\right) s + 4.4 \cdot 10^{-4}}{s^2 + 1.6\, s + 0.13} & 0 \end{pmatrix}$$

and $P^c(s)$

$$\begin{pmatrix} \frac{6.7 \cdot 10^{-7}}{s^2 + 1.5\, s + 3.2 \cdot 10^{-3}} & 0 & 0 \\ 0 & \frac{244.0}{s^2 + 1.5\, s + 3.3 \cdot 10^{-3}} & 0 \\ 0 & \frac{0.78}{s^2 + 1.6\, s + 0.13} & \frac{7.8}{s^2 + 1.6\, s + 0.13} \end{pmatrix}.$$

Notice that $Q^c(s)$ is no longer lower-triangular, but fully connected. Introducing loading effects creates additional coupling between nodes in the network. If the coupling is significant, the *designed* network interactions of the incoherent feedforward loop are overcome by the *crosstalk* network interactions [67], [56], [69], [70], [20], [71], [8]. Thus, the coupling that is introduced into the biochemical reaction network by loading effects is reflected in the structure of $(Q^c, P^c)(s)$.

In contrast, the transfer function of the crosstalk system only characterizes how system outputs causally depend on inputs. In particular, $G_c(s)$ is also a full matrix like $Q^c(s)$ of 6th order SISO transfer functions

$$G^c \equiv \begin{bmatrix} G^c_{11}(s) & G^c_{12}(s) & G^c_{13}(s) \\ G^c_{21}(s) & G^c_{22}(s) & G^c_{23}(s) \\ G^c_{31}(s) & G^c_{32}(s) & G^c_{33}(s) \end{bmatrix} (s)$$

but all structural information about how loading effects cause interference *among* system states is mixed with the information about how outputs causally depend on inputs in $G(s)$. An identification algorithm of entries in $G(s)$ will thus be *unable*

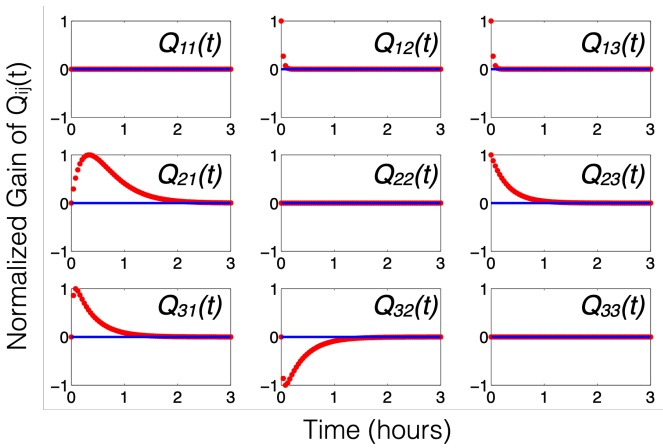

Fig. 3. **Dynamical structure functions describe how network structure evolves over time (and as a function of frequency):** The time-lapse response of the dynamical structure convolution kernel $Q^c(t) = \mathcal{L}^{-1}(Q^c(s))$ for the incoherent feedforward loop in system (7). By examining the functional response of each entry in $Q^c(t)$ (or $Q^c(s)$), we see that the network structure of the incoherent feedforward loop in Example II-A1 is a time-evolving, or *dynamic*, entity.

to quantify the size of crosstalk or interference among system states.

To what extent can the entries of $(Q(s), P(s))$ can be used to quantify the size of crosstalk in a synthetic gene networks? The following theorem shows that the dynamical structure function can be used to quantify crosstalk in biochemical reaction networks.

Theorem 1: Let $\mathcal{L}$ denote the two-sided Laplace operator. Suppose the states x^c and x^a of the systems (15) are (16) are shifted, so that the origin is a locally asymptotically stable equilibrium point and Q^c and Q^a are the respective dynamical structure functions calculated for each linearized system about the origin. Then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = Q^a_{ij}(s) - Q^c_{ij}(s) + \frac{\partial}{\partial Y_j}\mathcal{L}\left(O(x^2)\right) \tag{8}$$

and in particular, if

$$Q^a_{ij}(s) \equiv 0$$

then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = -Q^c_{ij}(s) + \frac{\partial}{\partial Y_j}\mathcal{L}\left(O(x^2)\right)$$

and can be estimated from input output data $(Y(s), U(s))$. The proof of this theorem is provided in the Supplementary Information. There are two ways to apply this result. First, if one has an idealized or reference dynamical structure model of the circuit, then this can be compared to the dynamical structure model estimated from the data. Second, in the absence of a reference model, the dynamical structure model can be estimated directly from data to observe new edge functions or gain mismatch directly from the data-driven dynamical structure model. In the latter scenario, if the crosstalk interactions are mediated on an existing edge, it will not be possible to separate the magnitude of the crosstalk from the existing (intended) dynamics of a given active edge in the network. However, the discovered edge dynamics can be compared to the intended behavior, e.g., intentional activation or repression over a certain growth phase of the cells.

III. Direct Estimation Algorithm for Dynamical Structure Functions

In this section, we will introduce a direct estimation algorithm for estimating the dynamical structure function. The dynamical structure function is a tuple of matrix transfer functions $(Q(s), P(s))$ and can be directly estimated from experimental or simulation data so long as the data and the conditions of the experiment satisfy the assumptions of the following theorem, proved in [62].

Theorem 2: Given a $p \times m$ transfer function $G(s)$, dynamical structure reconstruction is possible from partial structure information if and only if $p-1$ elements in each column of $[Q(s) \;\; P(s)]^*$ are known that uniquely specify the component of (Q, P) in the nullspace of $[G(s)^* \;\; I]$.

It follows from this theorem that without additional structural information about the columns of the matrix $[Q(s) \;\; P(s)]^*$, it is not possible to identify $(Q(s), P(s))$. In synthetic gene circuits with complex internal network interactions encoded by $Q(s)$, we can still solve for the structure and parameters of $Q(s)$, as long as enough elements of $P(s)$ are known. For example, targeted gene knockdowns (CRISPRi) or knockouts (engineered genomic mutations) can be engineered so that $P(s)$ is a diagonal matrix transfer function. That is, *all genes or biological states in the network of interest* are A) monitored by some measurement channel over time and B) independently perturbed by a diagonal element in $P(s)$. These are necessary and sufficient conditions for reconstruction of $Q(s)$ and $P(s)$ and align with the conditions developed by Sontag, Kholodenko, Wolkenhauer, Kang, Prabakaran et al. for network reconstruction of full state measurement systems [39], [40], [24], [41], [42], [43], [44], [45].

Under the above premises, the task is to estimate the diagonal transfer function entries of $P(s)$ and to estimate all off-diagonal entries of $Q(s)$. Recall from the derivation in Section II-A that the diagonal entries of $Q(s)$ are set to 0 by subtracting the diagonal entries of the precursor $W(s)$ transfer function matrix from $W(s)$. Further, by left-multiplying $(sI - D)^{-1}$ with $W(s) - D(s)$, this guarantees that the representation for $Y = QY + PU$ yields a unique $Q(s)$ and $P(s)$. The entries of $Q(s)$ and $P(s)$ are all strictly proper rational transfer functions and thus encode causal dynamics.

There are two approaches to solve for $Q(s)$ and $P(s)$. The first is to estimate the transfer function $G(s)$ using a standard transfer function estimation routine, followed by inversion of G and calculation of the entries of $P_{ii}(s)$ and subsequently the entries of $Q(s)$. This approach has the drawback of relying on inversion of the matrix transfer function $G(s)$, which often requires symbolic inversion and is thus prone to numerical instability and scaling issues for larger networks.

The second approach, which we propose here and develop code for, is to identify the dynamical structure function directly from data by writing the model estimation problem in normal form. First, we can estimate the discrete time approximations of $Q(s)$ and $P(s)$, as $\hat{Q}(z), \hat{P}(z)$, which will follow the same identifiability conditions [62].

Specifically, we have that

$$Y(z) = \hat{Q}(z)Y(z) + \hat{P}(z)U(z). \tag{9}$$

which given that $\hat{Q}$ and $\hat{P}$ share the same denominator, we can multiply the characteristic polynomial of $\hat{Q}(z)$ on both sides, to obtain

$$\left(z^{n_d} + a_1 z^{n_d-1} + \ldots + a_{n_d}\right) Y(z) = \hat{Q}_{num}(z)Y(z) + \hat{P}_{num}(z)U(z). \tag{10}$$

The matrices $\hat{Q}_{num}(z)$ contain unknown coefficients like $a_1, ..., a_{n_d}$ and so we can express the known quantity $z^{n_d}Y(z)$ after inverse Z-transforming to obtain

$$\begin{aligned} Y[t-n_d] = -a_1 Y[t-n_d+1] - a_2 Y[t-n_d+2] - \\ \ldots - a_{n_d} Y[t-n_d] - \Theta_{Q,num} Y_{t-m_d:t} \\ - \Theta_{P,num} Y_{t-m_d:t} \end{aligned} \tag{11}$$

which can be written in normal form as

$$b = x^T \Theta. \tag{12}$$

This equation can be stacked for multiple time traces, collected from different conditions or experimental replicates, or by staggering the time horizon, to obtain the stacked normal form equations

$$B = X\Theta. \tag{13}$$

where B and X are known values of time-series data and Θ contains the coefficients determining the characteristic polynomial and the numerators of the dynamical structure function. Once these estimates are obtained, a standard discrete-to-continuous transformation can be used to estimate $Q(s)$ and $P(s)$. The steps of our algorithm are summarized in Algorithm 1 and the full MATLAB code is provided at the Github repository `https://github.com/YeungRepo/NetRecon`.

A couple remarks are in order. First, there are two distinct hyperparameters that require optimization in generating the estimates for $Q(s)$ and $P(s)$: 1) n_d, the order of the characteristic polynomial and 2) the selection of the number of subsampled timepoints h_{max} in a given time-series traces. The optimal value of h_{max} will depend on the dataset, to ensure the condition number of the matrix X must be small. In high resolution time-series measurements, a small or short subsampled horizon h_{max} may produce virtually identical data if the transient has a slow rate of change, which can result in ill-conditioning of X. We optimize n_d, h_{max} to minimize the n-step L1 prediction error across all N_{exp} experimental samples. In practice, we observe that this criteria, by necessity, guarantees an appropriate selection of h_{max} as well as a lower optimal n_d value.

Secondly, the routine described above may appear to be easily cast as a classic transfer function estimation routine, similar to the tools developed by [73] in MATLAB. The primary difference in this procedure from a standard transfer function estimation procedure for

$$\begin{bmatrix} Q(z) & P(z) \end{bmatrix}$$

is that we must impose structural information about $Q(z)$ and $P(z)$ on the estimation process. This results in a structured

Algorithm 1: Direct QP Estimation Algorithm

Input: $Y_{Data}, U_{Data} \in \mathbb{R}^{N \times p \times N_{Exp}}$
Result: Dynamical structure function (Q(s),P(s))
Initialize:
$n_{d_{min}}$ (min. num. of poles),
$n_{d_{max}}$ (max. num. of poles),
h_{max} (max. horizon of subsampled time-series trace)
for $n_{ord} = n_{d_{min}}, , ..., n_{d_{max}}$ **do**
  while $1 < h_{max} < N - n_{ord} - 1$ **do**
    Initialize:
    $Y_{LH} = [\,]$ (stacked snapshots of time-series data corresponding to char. polynomial of $Q(s)$
    $Y_{RH} = [\,]$ (stacked snapshots of time-series data corresponding to numerator terms of Q)
    $U_{RH} = [\,]$ (stacked snapshots of time-series data corresponding to numerator terms of P)
    for $exp_{id} = 1, ...N_{exp}$ **do**
      for $c = 1, ...p$ **do**
        for $r = 1, ...h_{max}$ **do**
          $t_s = N - (r-1) - n_{ord}$
          $t_e = N - (r-1)$
          Y_{LH}.append($Y[t_s : t_e, c, exp_{id}]$)
          Initialize: Y_{alt} as empty array.
          for $y_{alt} = 1, .., p-1$ **do**
            if $y_{alt} + c > p$ **then**
              $c_{alt} = \text{mod}(y_{alt} + c), p)$
              else
              $c_{alt} = y_{alt} + c$
              end
            end
            $t_{s,alt} = N - (r-1) - n_{ord}$
            $t_{e,alt} = N - (r-1)$
            Y_{alt}.append($Y[t_{s,alt} : t_{e,alt}, c_{alt}, exp_{id}]$)
          end
          $z_{pre} = n_{ord}(c-1)(p-1)$
          $z_{post} = n_{ord}(y-c)(p-1)$
          Y_{RH}.append($\left[\mathbf{1}_{c\equiv 1}\mathbf{0}_{1\times z_{pre}} | Y_{alt} | \mathbf{1}_{c\equiv p}\mathbf{0}_{1\times z_{post}}\right]$)
          Initialize: U_{temp} as empty array.
          $t_s^u = N - r + 1 - n_{ord} + 1$
          $t_e^u = N - r + 1$
          for $u_{ind} = 1, ..., m$*(or p since $m = p$)* **do**
            U_{temp}.append($\mathbf{1}_{u_{ind}\equiv c} U_{Data}\,[t_s^u : t_e^u]$)
          end
          U_{RH}.append(U_{temp})
        end
      end
    end
    $X = [-Y_{LH}\,[:, 2 : n_{ord}+1] \,|\, Y_{RH} \,|\, U_{RH}]$
    $B = [Y_{LH}\,[:, 1]]$
    $\Theta(h_{max}) = X^{\dagger} B$
  end
end

system identification problem, which typically is non-convex and difficult to initialize properly. Specifically, the diagonal entries of the estimated $Q(z)$ and the off-diagonal entries of $P(z)$ must be exactly 0, as per the premises of [62]. Again, these structural constraints guarantee a unique representation of the dynamical structure function and identifiability of the model. When using standard transfer function estimation tools (in MATLAB's System Identification toolbox), we found repeatedly over thousands of numerical trials that imposing structural constraints as model priors resulted in A) models with extremely poor n-step prediction capacity (forecasting) or B) models with extremely poor 1-step fit scores. This motivated the development of a direct estimation algorithm, that mirrors the standard estimation of an discrete-time transfer function model, but where *structural constraints* of $Q(z)$ and $P(z)$ are directly encoded into the formulation of the normal form of equations (Lines 19-28, 32 and 36 of Algorithm 1).

IV. The Dynamical Structure of an *in vitro* Genelet Repressilator

We now illustrate the process of data-driven estimation of dynamical structure functions, using experimental data. In this section, we take as a first test case the synthetic genelet repressilator developed by Kim and Winfree [72]. The genelet repressilator consists of three DNA switches that repress one another through indirect sequestration. Specifically, each DNA switch transcribes its mRNA product only when its activator strand binds to complete its T7 RNA polymerase promoter sequence. The RNA product produced from each DNA switch, in turn, acts as an inhibitor to the downstream switch by binding to the downstream switch's DNA activator molecule. Thus, by sequestering the DNA activator from completing the T7 RNA polymerase promoter region, the mRNA product of the upstream switch inhibits activation of the downstream switch. Figure 4A shows the mechanistic design of the genelet switch.

The genelet switch relies heavily on RNase H to degrade any activator-mRNA inhibitor complexes. Without degradation, the binding of activator to mRNA inhibitor is much faster than unbinding and so sequestration is effectively irreversible. Thus, in order for the repressilator to function properly, RNase H must degrade its target substrates sufficiently fast. If RNase H is saturated with high levels of a particular substrate, this slows the degradation of other substrates, creating a crosstalk interaction between competing DNA-RNA complexes.

By performing network reconstruction on the genelet repressilator, we can determine how much crosstalk exists in the biocircuit. Further, we can validate our dynamical structure reconstruction algorithm in an *in vitro* setting, by *deliberately attenuating one of the components to create a gain imbalance*. We can see if the reconstruction process recovers the deliberate imbalance we introduce into the genelet repressilator, even when simply measuring local perturbations of an operating point for a normally oscillatory circuit.

To reconstruct $Q(s)$ and $P(s)$, we performed a single experiment with three perturbations applied in series [39], [40], [24], [41], [42], [43], [44], [45], [62]. To perturb each switch we pipetted a small perturbative concentration of DNA inhibitor (a DNA analogue of RNA inhibitor). Since DNA is not degradable in a T7 expression system by RNase H, it effectively acts as a step input since it binds to DNA activator and does not degrade. In this way, our perturbation design ensures sufficiency of excitation and independent perturbation of each activator (and downstream switch), thereby satisfying the identifiability conditions in [62] and the persistence of excitation conditions described in [73]. Further, we attenuated the concentration of the third switch T_{31} by 20%, to *create a deliberate gain imbalance for evaluating our reconstruction algorithm*.

A detailed model of the repressilator can be found in the supplement of [72]. Since the derivation is lengthy, it suffices to write the idealized dynamical structure function $Q^a(s)$ of this system, corresponding to the detailed model provided in Supplementary Section 1.6 [72]. The structure is obtained by linearizing the system, transforming into the Laplace domain, eliminating hidden variables to obtain the following:

$$\begin{bmatrix} 0 & 0 & Q^a_{13}(s) \\ Q^a_{21}(s) & 0 & 0 \\ 0 & Q^a_{32}(s) & 0 \end{bmatrix}$$

reflecting the cyclic structure of the system. This represents an idealized model of the system. As stated in Theorem **??** every entry where $Q^a_{ij}(s) \equiv 0$, the corresponding entry in $Q(s)$ estimated directly from experimental data, will be a crosstalk interaction present in the network. Here $Q(s)$ is used to denote $Q^c(s)$, the dynamical structure function estimated directly from data.

The experimental data used to fit $Q(s)$ and $P(s)$ are plotted in Figure 5, along with their respective fits. For each row i of $Q(s)$, we use $Y_j, j \neq i$ and U_i as inputs and Y_i as the output for a direct MIMO $p \times 1$ transfer function estimation problem. The impulse response for the convolution kernel $Q(t)$ of the reconstructed $Q(s)$ is plotted in Figure 6.

If we compute the corresponding $\mathcal{H}_\infty$ gain of each entry in $Q_{ij}(s)$ and scale by the maximum gain, we obtain

$$||Q(s)||_\infty = \begin{pmatrix} 0 & 0.07 & \mathbf{0.73} \\ \mathbf{1.0} & 0 & 0.3 \\ 0.053 & \mathbf{0.17} & 0 \end{pmatrix}.$$

We see significant crosstalk on the edge $Q_{23}(s)$ and minor crosstalk from entries $Q_{31}(s)$ and $Q_{12}(s)$. This crosstalk need not occur simultaneously, since the $\mathcal{H}_\infty$ gain calculates the worst-case or maximum gain over all possible frequencies. With the exception of $Q_{23}(s)$, all other crosstalk entries have strictly smaller $\mathcal{H}_\infty$ gain than the designed edge. Examining the impulse response of the convolution kernel confirms these observations; the crosstalk edge $Q_{23}(t)$ has a larger impulse response than *designed* edge $Q_{32}(t)$.

As intended in the design of the experiment, our estimated network model shows a gain imbalance between the *designed* edges $Q_{32}(s)$, $Q_{13}(s)$ and $Q_{21}(s)$. In order for the oscillator to perform properly, it needs to have approximately the same gain along each edge in the network. This example verifies that our reconstruction algorithm can identify important functional dynamics of a genetic circuit; especially for debugging pur-

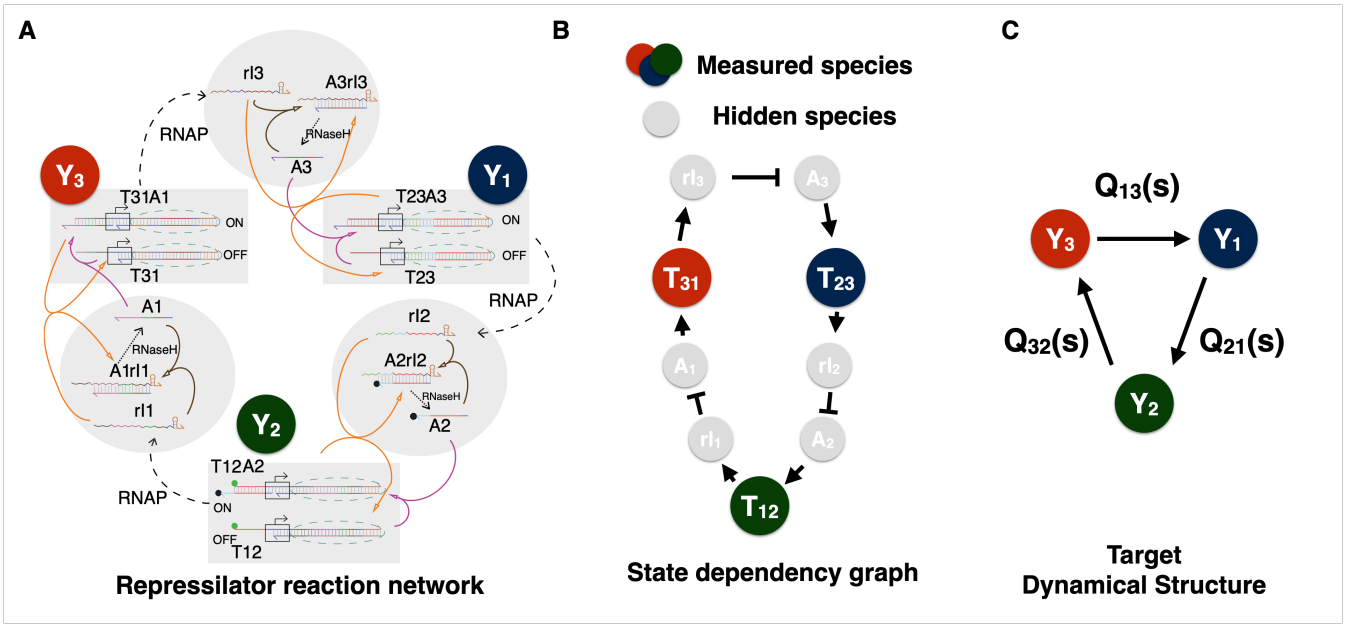

Fig. 4. **Network representations of a synthetic genelet repressilator: (A)** A reaction network using push-arrow reaction notation of the synthetic genelet repressilator. **(B)** A diagram representing the reaction dynamics in panel **(A)** as state dependencies from a nonlinear ODE model in [72]. **(C)** The dynamical structure of the repressilator (without inputs), with nodes representing measured chemical species and edge weights corresponding to entries in $Q(s)$.

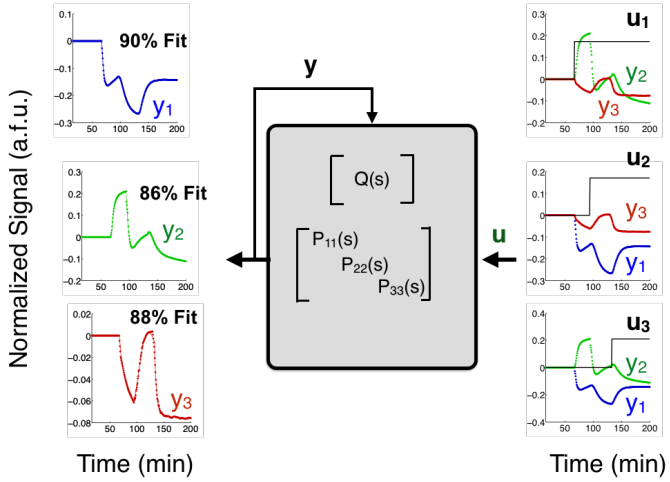

Fig. 5. Time-series experimental data from *in vitro* network perturbation experiments of a T7 RNAP genelet repressilator. Three outputs are measured simultaneously, y_1, y_2, and y_3, corresponding to DNA switches T_{31}, T_{12} and T_{23}. DNA homologues of the RNA inhibitors rI_j $j = 1, 2, 3$ are injected at small concentrations to provide a step input perturbation to the corresponding component Y_j in the genelet circuit.

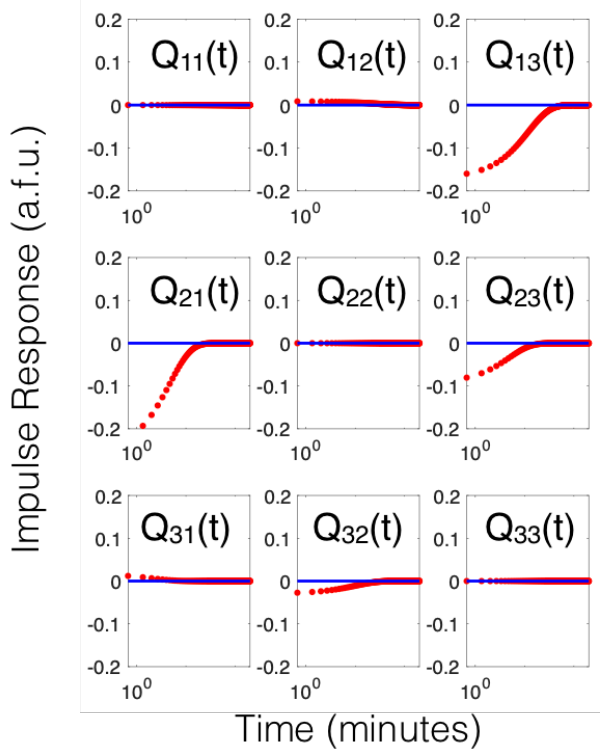

Fig. 6. Impulse response of the estimated convolution kernel $Q(t)$ matrix. $Q(s)$ is estimated directly from experimental data, transformed into the frequency domain, and simulated in time for $t = 0$ to $t = 300$ minutes.

poses. The linearization scheme is valid, so long as we model fluctuations in dynamics from a nominal initial condition, even if the initial condition is not stable or leads to oscillatory dynamics Our results here illustrate how linearized models can provide insight into local dynamics. In this simple, controlled dataset, we know we can increase the gain of the edge in $Q_{32}(s)$ by adjusting the binding affinity of the activator DNA with its inhibitor RNA, or by increasing the concentration of the corresponding downstream switch T_{31}. Notice that this design insight may not be obvious by direct examination of experimental trajectories of each switch in Figure 5. As long as we have an idealized network model, we can measure the deviation from that model $Q^a(s)$ in the network model identified from the data $Q^c = Q(s)$ and identify edges or nodes in our network that need tuning.

V. The Dynamical Structure of an *in vivo* Transcriptional Event Detector

We now introduce a new transcriptional event detector circuit, one that is designed, built, and constructed for illustrating the use of our dynamical structure estimation algorithm in *in vivo* circuit design. Event detectors are useful because of their ability to perform temporal logic. Making temporal logic decisions enable applications such as programmed differentiation, where the goal is to perform some operation based on a combinatorial and temporal sequences of events that dictate cell fate.

So far there are two demonstrations of temporal logic gates: 1) a temporal logic gate that differentiates start times of two chemical outputs [76] and 2) a molecular counter that counts the number of sequential pulses of inducers [77]. Both event detectors use serine integrases to perform irreversible recombination, while [77] demonstrates the use of transcription-based event detecting to perform event counting. The advantage of an integrase-based approach is the persistent nature of DNA-based memory. At the same time, the drawback of integrase-based event detection is that it is limited to one-time use.

In contrast, transcription based event detectors use proteins instead of DNA to encode a memory state [77], [78]. The advantage of a transcription-based event detector is that proteins are labile, since they are diluted through cell growth or can be tagged for degradation. Thus, a transcriptional event detector's memory state can be reset after some period of time. On the other hand, maintaining protein state over multiple generations is metabolically expensive [71] and the dynamics of the circuit can become sensitive to production and growth phase of the cells. Therefore, a transcription based event detector biocircuit must be designed with precise timing, balance of production rates, and carefully tuned gain of each transcriptional regulator. This presents a suitable application for our network reconstruction algorithm.

A. Designing a transcriptional event detector

We designed our transcriptional event detector to be made of two constitutively expressed relay genes, AraC and LasR, and an internal toggle switch. The two relay genes transmit the arrival of two distinct induction events (arabinose and HSL) to relay output promoters pBAD and pLas respectively, which drive production of a fluorescent response in two relay promoters. To record these induction events historically, the output of each relay gene is coupled to one of two combinatorial promoters (pBAD-Lac or pLas-Tet) in a toggle switch. Each combinatorial promoter implements NIMPLY logic, e.g. pBAD-Lac (pLas-Tet) expresses TetR (LacI) only when arabinose (HSL) and AraC (LasR) are present and LacI (TetR) is absent. Thus, when one analyte (e.g. arabinose) arrives, it triggers latching of the toggle switch only if the toggle switch is unlatched to begin with or the prior latching protein state has been diluted out. The relay outputs thus transmit the *current or recent* induction event state while the toggle switch maintains the *historical* induction event state. Depending on the order of arrival of each inducer, we obtain different biocircuit states. Figure 7 details the genetic elements in the event detector biocircuit and the designed component interaction network.

We can write down an idealized model for the event detector (assuming no crosstalk), assuming first order degradation and production, with Hill functions encoding the NIMPLY logic of each promoter in the memory module.

$$\begin{aligned}
\dot{x}_1 &= \rho_1 m_1 - \delta_p x_1, \\
\dot{x}_2 &= \rho_2 m_2 - \delta_p x_2, \\
\dot{x}_3 &= \rho_3 m_3 - \delta_p x_3, \\
\dot{x}_4 &= \rho_4 m_4 - \delta_p x_4, \\
\dot{m}_1 &= \frac{k_1(k_l + u_1/k_{M,u1})}{(1 + u_1/k_{M,u1})} - \delta_m m_1, \\
\dot{m}_2 &= \frac{k_2(k_l + u_1/k_{M,u1})}{(1 + x_3/k_{M,3} + u_1/k_{M,u1})} - \delta_m m_2, \\
\dot{m}_3 &= \frac{k_3(k_l + u_2/k_{M,u2})}{(1 + x_2/k_{M,2} + u_2/k_{M,u2})} - \delta_m m_3, \\
\dot{m}_4 &= \frac{k_4(k_l + u_2/k_{M,u2})}{(1 + u_2/k_{M,u2})} - \delta_m m_4, \\
\mathbf{y} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^T & \mathbf{m}^T \end{bmatrix}^T
\end{aligned} \tag{14}$$

where the measured outputs of the system are $y_i = x_i, i = 2, 3$, ρ_i is the translation rate of m_i into x_i, δ_p is the effective dilution rate of $x_i, i = 1, ..., 4$, δ_m is the combined dilution and degradation rate of $m_i, i = 1, ..., 4$, k_M, u_i is the Michaelis constant for u_i, k_l is the leaky catalytic transcription rate, k_i is the catalytic transcription rate for m_i, and u_1, u_2 are arabinose and HSL, respectively.

Again, the dynamical structure function for this system is calculated by linearizing the system about a nominal initial condition, (x_0, m_0), taking a Laplace transform and solving out the hidden variables $m_1, ..., m_4$. We present a simplified case here, assuming algebraic symmetry of the parameters $k_i = k, \rho_i = \rho, k_{M,i} = k_M$ as it does not qualitatively change the structure of $(Q(s), P(s))$. We obtain:

$$Q^a(s) = \begin{bmatrix} 0 & Q_{12}(s) \\ Q_{21}(s) & 0 \end{bmatrix}$$

$$P^a(s) = \begin{bmatrix} P_{11}(s) & 0 \\ 0 & P_{22}(s) \end{bmatrix}$$

where $P_{ii}(s) = \rho/(\delta_m + s)(\delta_p + s)$ for $i = 1, 2$ and

$$\begin{aligned}
Q_{12}(s) &= \frac{-k\rho(k_l + u_1/k_M)}{k_M(\delta_m + s)(\delta_p + s)(u_1/k_M + x_3/k_M + 1)^2} \\
Q_{21}(s) &= \frac{-k\rho(k_l + u_2/k_M)}{k_M(\delta_m + s)(\delta_p + s)(u_2/k_M + x_2/k_M + 1)^2}
\end{aligned}$$

In the absence of protein degradation, $Q_{12}(s)$ and $Q_{21}(s)$ can be approximated with first-order SISO transfer functions. These expressions for $Q(s)$ and $P(s)$ are for the idealized dynamical structure function of the *alternative* system. Notice that $Q_{12}(s)$ and $Q_{21}(s)$ are strictly negative transfer functions, indicating the repression present in an idealized simulation of the event detector circuit. This is the intended *dynamical*

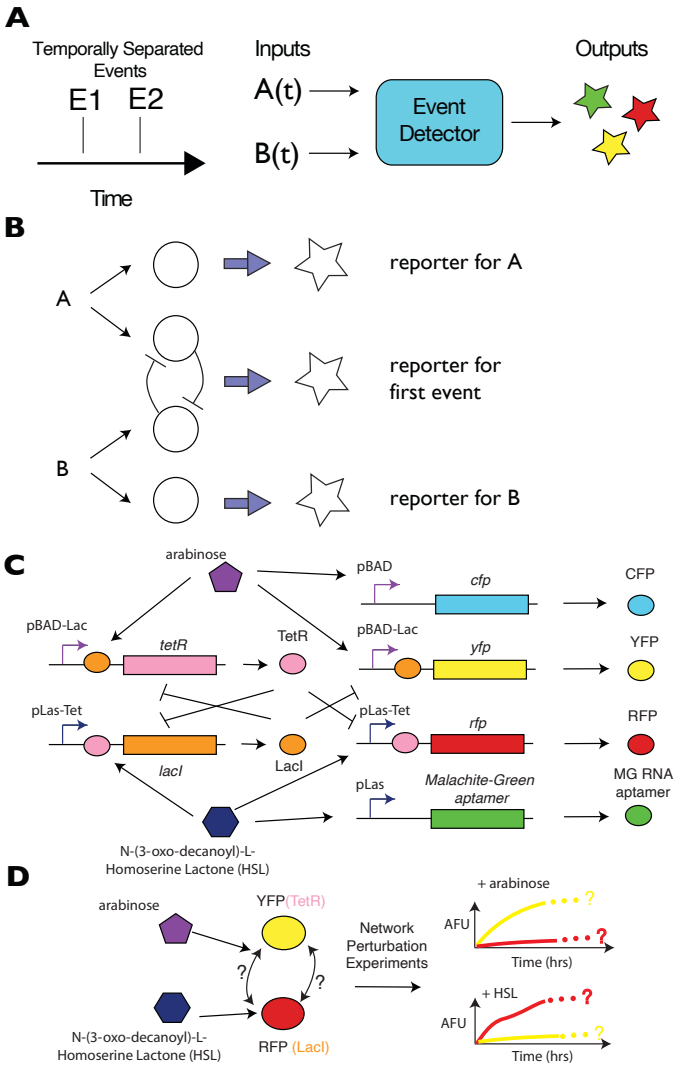

Fig. 7. (**A**) Left: We design an event detector to determine the identity and relative ordering of two events E_1 and E_2 occurring within a finite time horizon. (**B**) A schematic showing the logic of the circuit for the event detector. Arrival of event type A triggers transient reporter for A (top) and latching of the toggle in a A-dominant state as a memory state. Similarly, arrival of event type B triggers transient reporter for B (bottom) and latching of the toggle in a B-dominant state as a memory state. (**C**) A diagram showing the synthetic biocircuit parts used to implement the network architecture in (**B**). **D**) The arabinose and HSL inducers independently perturb distinct elements of the memory module in the event detector; a network model of the dynamic graph of the event detector can be reconstructed using dynamical structure function reconstruction experiments.

network structure of the event detector, in the absence of all genetic crosstalk or context effects.

Depending on the abundance of transcription factors such as LacI, TetR, and AraC, as well as commonly shared transcriptional and translational proteins, the *actual* dynamical structure function $Q^c(s)$ may not exhibit monotonic repression or may even unveil unwanted interactions. We can investigate these interactions under a range of conditions with dynamical structure estimation.

We constructed a biological implementation of the event detector, using the design specified in Figure 7. The logical components containing the relays and the memory module were encoded on to a plasmid vector with a kanamycin resistance marker and a ColE1 (high copy) replication origin. The fluorescent reporter elements with the relay promoters and readouts for the toggle switch were encoded on a plasmid vector with chloramphenicol resistance and the p15 replication origin.

B. Event Detector Latching Experiments

We evaluated the performance of our transcriptional event detector circuit using a temporal logic test. A standard temporal logic experiment for any two-input event detector is to evaluate the effect of varying the order of presentation of two input signals. In one test, we present the first input, arabinose, for 7.5 hours, followed by induction of the second input, a homo-serine lactone (HSL) quorum sensing molecule to activate the pLas-Tet promoter. In the second test, we swap the order of the inputs, presenting HSL quorum sensing molecule to the event detector for 7.5 hours, then present arabinose inducer as a second input. Both tests evaluate the ability of the memory module of the event detector to latch in the correct state in response to the first input, followed by a challenge to ignore the second input signal while the relays detect and read out the second input signal. The data for both of these *in vivo* tests is plotted in Figure 8B-C.

The event detector showed the correct latching response in all tests at standard maximum induction concentrations of arabinose (1 mM) and working induction concentrations of 1 μM HSL. For example, Figure 8C shows that when the event detector is given arabinose followed by HSL, it generates the correct fluorescent response of yellow fluorescent protein, with lower expressions level of RFP. Conversely, when we add HSL first, followed by arabinose, RFP signal ramps up immediately beginning as early as 1-2 hours after induction while YFP expression is abolished to background levels.

We tested a variety of combinations of high and low concentrations for arabinose and HSL. When the concentration of HSL was decreased to 1 nM, we observed consistent leaks in the memory module in either the YFP channel or the RFP channel. Decreasing arabinose down to 1 μM still allows for latching of high YFP expression, but in the presence of 1 μM HSL, any arabinose latching is reversed by HSL induction (data not plotted). Conversely, when we attenuate HSL induction to 1 nM, HSL does not prevent arabinose from reversing a HSL latch on the the memory module, see Figure 8B. This leak is significant enough in the 1 nM HSL induction level that the difference in signal between the arabinose-HSL induction scenario versus the HSL-arabinose induction scenario vanished. This temporal logic response profile is evident of a glitch in the event detector circuit that occurs at lower HSL and arabinose concentrations.

C. Network Reconstruction Experiments to Debug Circuit Failure

We conducted 4 *in vivo* network reconstruction experiments (2 inducers versus 2 concentrations), recording time-series data of the memory module relay elements, YFP and RFP. The memory module is designed using two hybrid promoters, so from a design standpoint, verification of the memory module

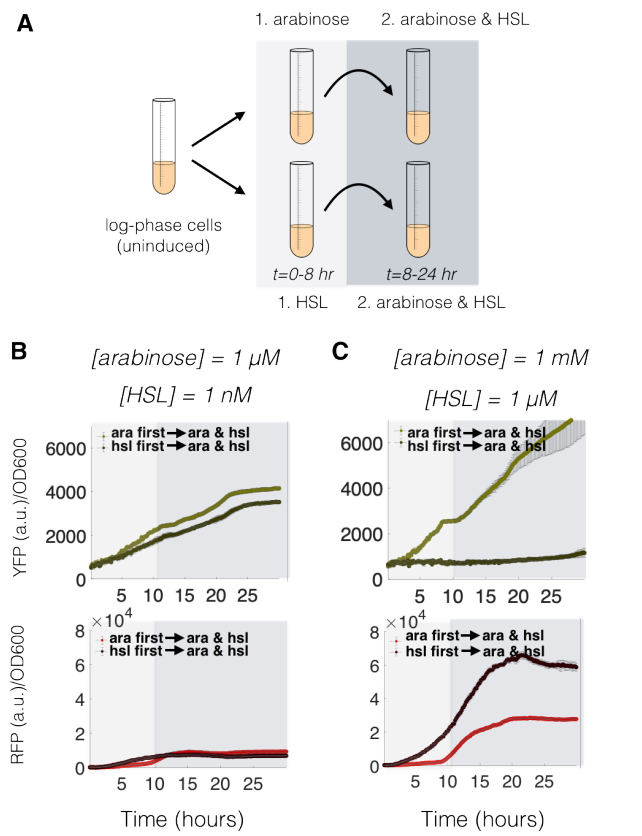

Fig. 8. A plot of data from *in vivo* plate reader experiments, testing the temporal logic properties of the event detector diagrammed in Figure 8. Notice that at 1 μM HSL and $1mM$ arabinose, the event detector functions properly, expressing different levels of YFP and RFP depending on the the order of arrival of arabinose and HSL. At 1 nM HSL and 1 μM arabinose induction concentrations, the temporal logic properties of the event detector completely are abolished.

was most critical. The arabinose inducer targets the pAra-Lac promoter, while the HSL inducer targets the pLas-Tet promoter (see Supplementary Information for sequences).

As shown in the model (14) of the event detector, the actual event detector we constructed exhibits nonlinear response. However, for any one parametric concentration regime, e.g. at a fixed arabinose or HSL concentration, the response of the system behaves similar to that of a linear system. Thus, we estimated a dynamical structure function for both conditions of the reconstruction experiment. The one-step accuracy in fitting dynamical structure models to the low gain condition (1 μM arabinose, 1 nM HSL) and high gain condition (1 mM arabinose and 1 μM HSL) were 99.996% and 99.995% respectively.

As in the case of the genelet repressilator, we can plot a dynamical network graph for the *in vivo* event detector to understand how the memory module components labeled by YFP and RFP, representing TetR and LacI respectively, interact with each other. A movie visualizing the dynamics of the edges of the graph is available for download (see Supplementary Information). Each edge represents the convolution kernel response of the edge to an impulse applied to that input. All responses are superimposed to form a dynamical graph. Snapshots of the graph are plotted in Figure 10, while time-lapse responses of the weights of each edge are plotted in Figure 9. Again as with the repressilator, we can see that the regulatory nature of edges in the event detector's memory module manifests as two edges with negative or positive values indicating repression or activation, respectively.

The reconstructed network of our transcriptional event detector reveals the functional relationship between states in the circuit at different concentration regimes. At lower concentrations of arabinose and HSL, the reconstructed transcriptional event detector network reveals functional cause of failed circuit latching. Both edges in the memory module did not repress their target promoters as intended, while the pLas-Tet promoter appears to enact a much higher gain of activated expression from HSL induction than does the activated expression of the pAra-Lac promoter in response to arabinose.

In the high gain setting, where arabinose is induced at 1 mM and HSL is induced at 1 μM, we see that the memory module exhibits the proper mutually repressing motif characteristic of the genetic toggle switch up after the arrival of the HSL inducer. The repression in both edges steps up their gain as t approaches 4 hrs, which is roughly the time when we see a plateauing of production in the RFP signal in Figure 8C. From our reconstruction model, we can see that the edges are well balanced at the higher concentration of inducers. At the low gain of inducers, the network is completely inactive, even though the genetic sequence of the circuit is the same. This example shows that our network verification algorithm can be used to determine the conditions, or the performance envelope, under which the circuit is functioning properly. Even though the underlying model of our system is a linear approximation to a nonlinear system, we can test the system at multiple initial conditions, operating points, or equilibria, to quantify network behavior of the system locally. Taken in aggregate, these can provide a parameterized view of how the network behaves over a range of experimental conditions.

VI. Conclusion

The dynamical structure function models the dependencies among measured states. It is a flexible representation of network structure that naturally adapts to the constraints imposed by experimental measurement. Since identifiability conditions of the dynamical structure function have been well characterized, appropriate experimental design can ensure that the process of network reconstruction produces a sensible answer.

In this work, we introduced a network reconstruction algorithm and a code base for reconstructing the dynamical structure function from data, to enable discovery and visualization of graphical relationships in a genetic circuit diagram as *time-dependent functions* rather than static, unknown weights. We proved a theorem, showing that dynamical structure functions can provide a data-driven estimate of the size of crosstalk fluctuations from an idealized model. We then illustrated these findings with numerical examples. Next, we used an *in vitro* genetic circuit, deliberately tuned with gain imbalance, to validate our algorithm on experimental data. Finally, we built

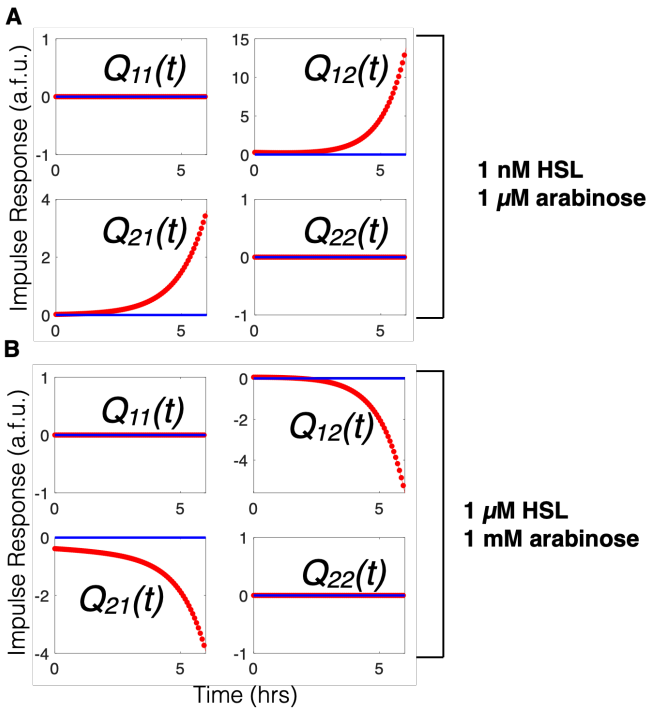

Fig. 9. Impulse response of the estimated convolution kernel $Q(t)$ matrix when the event detector biocircuit is induced with (A) 1 nM HSL and 1 μM arabinose or (B) 1 μM HSL and 1 mM arabinose. $Q(s)$ is estimated directly from experimental data, transformed into the frequency domain, and simulated in time as a function of hours from arrival time of an inducer input.

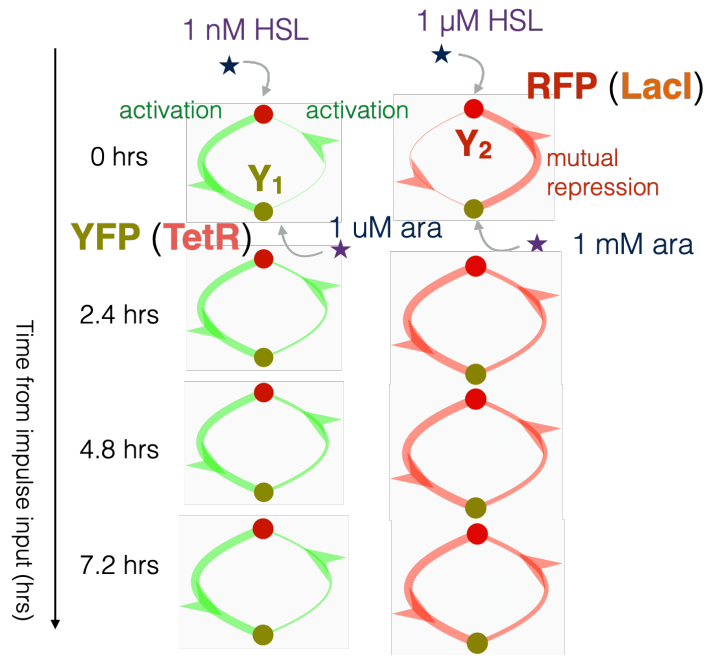

Fig. 10. A visualization of the impulse response of the estimated convolution kernel $Q(t)$ matrix when the event detector biocircuit is induced with low (left) versus high (right) concentrations of arabinose and HSL inducer. The width of edges in this graph coincide with the magnitude of the impulse response, while coloring is red if the sign of the impulse response for a given edge is negative (repression) and green if the given edge is positive (activation).

a new *E. coli* based transcriptional event detector and showed how estimation of the dynamical structure reveals active and inactive network states, depending on inducer concentration. These results show how the dynamical structure function characterizes the operational or active network. They also provide a route for future study of relationships between environmental parameters, active network dynamics, and biocircuit performance.

VII. Experimental Methods

All plasmids were constructed using either Golden Gate assembly [79] or Gibson isothermal assembly [80] in *E. coli*. Plasmids were sequence verified in JM109 cloning strains and transformed into the strain MG1655ΔLacI, provided as a courtesy by R. J. Krom and J. J. Collins. The event detector was transformed as a two-plasmid system with kanamycin and chloramphenicol selection. All *in vivo* experiments were carried out with $n = 2$ replicates using MatriPlates (Brook Life Science Systems MGB096-1-2-LG-L) 96 square-well glass bottom plates at 29° C in a H1 Synergy Biotek plate reader using 505/535 nm and 580/610 nm excitation/emission wavelengths. Cell density was quantified with optical density at 600 nm.

For *in vitro* experiments, all genelet repressilator reconstruction experiments were carried out at 37° C in a Horiba Spectrofluoremeter with 1 minute readout times, using Rhodamine Green, TYE 563 and Texas Red flourophores with 10 nm monochromator excitation and emission bands centered at 502/527, 549/563, and 585/615 nm respectively. All event detector network reconstruction reactions were performed using 500 μL reaction volumes in transformed *E. coli*, grown in square well glass-bottom plates using MatriPlates (Brook Life Science Systems MGB095-1-2-LG-L) with Luria-Bertain rich media broth at 29° C.

VIII. Author Contributions

E. Y. wrote the paper. E.Y, J.G., Y.Y., J. K., and R. M. M. edited drafts of the paper. E.Y. and J. K. designed and carried out experiments and processed experimental data. E.Y. performed analysis and modeling. J. G. and R. M. M. secured research funding. R. M. M. supervised the research process.

IX. Acknowledgments

We would like to acknowledge Sean Warnick, Vipul Singhal, Shara Balakrishnan, and Anandh Swaminanthan for insightful conversations on network reconstruction algorithms. We would like to thank and acknowledge Victoria Hsiao, Ophelia Venturelli, Clarmyra Hayes, Emmanuel de los Santos, Joe Meyerowitz, and Zachary Sun for guidance with experimental techniques. This work was supported by the Engineering and Physical Sciences Research Council, the Luxembourg National Research Foundation, Air Force Office of Scientific Research, Grant FA9550-14-1-0060, the Defense Advanced Research Projects Agency, Grants HR0011-12-C-0065 and FA8750-19-2-0502, the Army Research Office Young Investigator Program, Grant W911NF-20-1-0165, the

National Science Foundation, Grant 1317291, and the John and Ursula Kanel Charitable Foundation.

References

[1] J. A. Brophy and C. A. Voigt, "Principles of genetic circuit design," *Nature methods*, vol. 11, no. 5, pp. 508–520, 2014.

[2] E. Franco, E. Friedrichs, J. Kim, R. Jungmann, R. Murray, E. Winfree, and F. C. Simmel, "Timing molecular motion and production with a synthetic transcriptional clock," *Proceedings of the National Academy of Sciences*, vol. 108, no. 40, pp. E784–E793, 2011.

[3] K. Endo, K. Hayashi, T. Inoue, and H. Saito, "A versatile cis-acting inverter module for synthetic translational switches," *Nature communications*, vol. 4, no. 1, pp. 1–9, 2013.

[4] J. Hsia, W. J. Holtz, M. M. Maharbiz, M. Arcak, and J. D. Keasling, "Modular synthetic inverters from zinc finger proteins and small rnas," *PloS one*, vol. 11, no. 2, p. e0149483, 2016.

[5] R. Weiss, S. Basu, S. Hooshangi, A. Kalmbach, D. Karig, R. Mehreja, and I. Netravali, "Genetic circuit building blocks for cellular computation, communications, and signal processing," *Natural Computing*, vol. 2, no. 1, pp. 47–84, 2003.

[6] A. Tamsir, J. J. Tabor, and C. A. Voigt, "Robust multicellular computing using genetically encoded nor gates and chemical ?wires?" *Nature*, vol. 469, no. 7329, pp. 212–215, 2011.

[7] D. Del Vecchio, A. J. Ninfa, and E. D. Sontag, "Modular cell biology: retroactivity and insulation," *Molecular Systems Biology*, vol. 4, no. 1, p. 161, 2008.

[8] A. Gyorgy and D. Del Vecchio, "Retroactivity to the input in complex gene transcription networks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 3595–3601.

[9] S. Jayanthi and D. D. Vecchio, "On the compromise between retroactivity attenuation and noise amplification in gene regulatory networks," *Proceedings of the 2009 IEEE Conference on Decision and Control*, December 2009.

[10] S. Zhang and C. A. Voigt, "Engineered dcas9 with reduced toxicity in bacteria: implications for genetic circuit design," *Nucleic acids research*, vol. 46, no. 20, pp. 11 115–11 125, 2018.

[11] Y. Zhao, L. Li, G. Zheng, W. Jiang, Z. Deng, Z. Wang, and Y. Lu, "Crispr/dcas9-mediated multiplex gene repression in streptomyces," *Biotechnology journal*, vol. 13, no. 9, p. 1800121, 2018.

[12] X. Zhang, J. Wang, Q. Cheng, X. Zheng, G. Zhao, and J. Wang, "Multiplex gene regulation by crispr-ddcpf1," *Cell discovery*, vol. 3, no. 1, pp. 1–9, 2017.

[13] C. A. Hutchison, R.-Y. Chuang, V. N. Noskov, N. Assad-Garcia, T. J. Deerinck, M. H. Ellisman, J. Gill, K. Kannan, B. J. Karas, L. Ma *et al.*, "Design and synthesis of a minimal bacterial genome," *Science*, vol. 351, no. 6280, 2016.

[14] A. A. Nielsen and C. A. Voigt, "Multi-input crispr/c as genetic circuits that interface host regulatory networks," *Molecular systems biology*, vol. 10, no. 11, p. 763, 2014.

[15] Z. Hou, P. Jiang, S. A. Swanson, A. L. Elwell, B. K. S. Nguyen, J. M. Bolin, R. Stewart, and J. A. Thomson, "A cost-effective rna sequencing protocol for large-scale gene expression studies," *Scientific reports*, vol. 5, p. 9570, 2015.

[16] M. W. van der Kamp, R. D. Schaeffer, A. L. Jonsson, A. D. Scouras, A. M. Simms, R. D. Toofanny, N. C. Benson, P. C. Anderson, E. D. Merkley, S. Rysavy *et al.*, "Dynameomics: a comprehensive database of protein dynamics," *Structure*, vol. 18, no. 4, pp. 423–435, 2010.

[17] H. Link, T. Fuhrer, L. Gerosa, N. Zamboni, and U. Sauer, "Real-time metabolome profiling of the metabolic switch between starvation and growth," *Nature Methods*, vol. 12, no. 11, pp. 1091–1097, 2015.

[18] J. R. Karr, J. C. Sanghvi, D. N. Macklin, M. V. Gutschow, J. M. Jacobs, B. Bolival Jr, N. Assad-Garcia, J. I. Glass, and M. W. Covert, "A whole-cell computational model predicts phenotype from genotype," *Cell*, vol. 150, no. 2, pp. 389–401, 2012.

[19] A. Schöllig, U. Münz, and F. Allgöwer, "Topology-dependent stability of a network of dynamical systems with communication delays," in *2007 European Control Conference (ECC)*. IEEE, 2007, pp. 1197–1202.

[20] A. Gyorgy, "Sharing resources can lead to monostability in a network of bistable toggle switches," *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 308–313, 2018.

[21] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*. Springer Science & Business Media, 2013, vol. 36.

[22] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick, "Representing structure in linear interconnected dynamical systems," *The Proceedings of the 49th IEEE Conference on Decision and Control, to appear*, December 2010.

[23] L. Sommerlade, M. Eichler, M. Jachan, K. Henschel, J. Timmer, and B. Schelter, "Estimating causal dependencies in networks of nonlinear stochastic dynamical systems," *Physical Review E*, vol. 80, no. 5, p. 051128, 2009.

[24] T. Kang, R. Moore, Y. Li, E. Sontag, and L. Bleris, "Discriminating direct and indirect connectivities in biological networks," *Proceedings of the National Academy of Sciences*, vol. 112, no. 41, pp. 12 893–12 898, 2015.

[25] A. J. Gates and L. M. Rocha, "Control of complex networks requires both structure and dynamics," *arXiv preprint arXiv:1509.08409*, 2015.

[26] D. M. Wolf and A. P. Arkin, "Motifs, modules and games in bacteria," *Current opinion in microbiology*, vol. 6, no. 2, pp. 125–134, 2003.

[27] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proceedings of the National Academy of Sciences*, vol. 100, no. 21, pp. 11 980–11 985, 2003. [Online]. Available: http://www.pnas.org/content/100/21/11980.abstract

[28] S. Cardinale and A. Arkin, "Contextualizing context for synthetic biology–identifying causes of failure of synthetic biological systems," *Biotechnology Journal*, 2012.

[29] P. Fontanarrosa, H. Doosthosseini, A. Espah Borujeni, Y. Dorfan, C. A. Voigt, and C. J. Myers, "Genetic circuit dynamics: Hazard and glitch analysis," *ACS Synthetic Biology*, 2020.

[30] J. J. Tyson, K. C. Chen, and B. Novak, "Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell," *Current opinion in cell biology*, vol. 15, no. 2, pp. 221–231, 2003.

[31] L. Goentoro, O. Shoval, M. W. Kirschner, and U. Alon, "The incoherent feedforward loop can provide fold-change detection in gene regulation," *Molecular cell*, vol. 36, no. 5, pp. 894–899, 2009.

[32] L. Goentoro and M. W. Kirschner, "Evidence that fold-change, and not absolute level, of β-catenin dictates wnt signaling," *Molecular cell*, vol. 36, no. 5, pp. 872–884, 2009.

[33] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, 2000.

[34] H. El Samad, D. Del Vecchio, and M. Khammash, "Repressilators and promotilators: Loop dynamics in synthetic gene networks," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 4405–4410.

[35] J. Garcia-Ojalvo, M. B. Elowitz, and S. H. Strogatz, "Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing," *Proceedings of the National Academy of Sciences*, vol. 101, no. 30, pp. 10 955–10 960, 2004.

[36] S. Müller, J. Hofbauer, L. Endler, C. Flamm, S. Widder, and P. Schuster, "A generalized model of the repressilator," *Journal of mathematical biology*, vol. 53, no. 6, pp. 905–937, 2006.

[37] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Letters to Nature*, vol. 403, January 2000.

[38] L. Marucci, D. A. Barton, I. Cantone, M. A. Ricci, M. P. Cosma, S. Santini, D. di Bernardo, and M. di Bernardo, "How to turn a genetic circuit into a synthetic tunable oscillator, or a bistable switch," *PloS one*, vol. 4, no. 12, p. e8083, 2009.

[39] O. Wolkenhauer, P. Wellstead, K.-H. Cho, and E. D. Sontag, "Network reconstruction based on steady-state data," *Essays in Biochemistry*, vol. 45, pp. 161–176, 2008.

[40] S. Prabakaran, J. Gunawardena, and E. Sontag, "Paradoxical results in perturbation-based signaling network reconstruction," *Biophysical journal*, vol. 106, no. 12, pp. 2720–2728, 2014.

[41] E. Sontag, A. Kiyatkin, and B. N. Kholodenko, "Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data," *Bioinformatics*, vol. 20, no. 12, pp. 1877–1886, 2004.

[42] T. Quarton, T. Kang, E. D. Sontag, and L. Bleris, "Exploring the impact of resource limitations on gene network reconstruction," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3350–3355.

[43] T. Kang, J. T. White, Z. Xie, Y. Benenson, E. Sontag, and L. Bleris, "Reverse engineering validation using a benchmark synthetic gene circuit in human cells," *ACS synthetic biology*, vol. 2, no. 5, pp. 255–262, 2013.

[44] S. Wang, J.-R. Lin, E. D. Sontag, and P. K. Sorger, "Inferring reaction network structure from single-cell, multiplex data, using toric systems theory," *PLoS computational biology*, vol. 15, no. 12, p. e1007311, 2019.

[45] M. Andrec, B. Kholodenko, R. Levy, and E. Sontag, "Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy," *J. Theoret. Biol.*, vol. 232, no. 3, pp. 427–441, 2005.

[46] B. N. Kholodenko, A. Kiyatkin, F. J. Bruggeman, E. Sontag, H. V. Westerhoff, and J. B. Hoek, "Untangling the wires: a strategy to trace functional interactions in signaling and gene networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 20, pp. 12 841–12 846, 2002.

[47] R. Schuetz, L. Kuepfer, and U. Sauer, "Systematic evaluation of objective functions for predicting intracellular fluxes in escherichia coli," *Molecular systems biology*, vol. 3, no. 1, p. 119, 2007.

[48] A. Hilfinger, T. M. Norman, G. Vinnicombe, and J. Paulsson, "Constraints on fluctuations in sparsely characterized biological systems," *Physical review letters*, vol. 116, no. 5, p. 058101, 2016.

[49] A. Hilfinger, T. M. Norman, and J. Paulsson, "Exploiting natural fluctuations to identify kinetic mechanisms in sparsely characterized systems," *Cell systems*, vol. 2, no. 4, pp. 251–259, 2016.

[50] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[51] A. K. Tangirala, *Principles of system identification: theory and practice*. Crc Press, 2018.

[52] E. Yeung, J. L. Beck, and R. M. Murray, "Modeling environmental disturbances with the chemical master equation," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 1384–1391.

[53] E. Yeung, J. Kim, and R. M. Murray, "Resource competition as a source of non-minimum phase behavior in transcription-translation systems," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 4060–4067.

[54] A. Gyorgy, J. I. Jiménez, J. Yazbek, H.-H. Huang, H. Chung, R. Weiss, and D. Del Vecchio, "Isocost lines describe the cellular economy of genetic circuits," *Biophysical journal*, vol. 109, no. 3, pp. 639–646, 2015.

[55] A. Gyorgy and D. Del Vecchio, "Limitations and trade-offs in gene expression due to competition for shared cellular resources," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 5431–5436.

[56] Y. Qian, H.-H. Huang, J. I. Jiménez, and D. Del Vecchio, "Resource competition shapes the response of genetic circuits," *ACS synthetic biology*, vol. 6, no. 7, pp. 1263–1272, 2017.

[57] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.

[58] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, no. 3, pp. 579–616, 1991.

[59] A. Reynolds, D. Leake, Q. Boese, S. Scaringe, W. S. Marshall, and A. Khvorova, "Rational sirna design for rna interference," *Nature biotechnology*, vol. 22, no. 3, pp. 326–330, 2004.

[60] J. R. Elmore, G. N. Dexter, R. Francis, L. Riley, J. Huenemann, H. Baldino, A. M. Guss, and R. Egbert, "The sage genetic toolkit enables highly efficient, iterative site-specific genome engineering in bacteria," *bioRxiv*, 2020.

[61] R. Lutz and H. Bujard, "Independent and tight regulation of transcriptional units in escherichia coli via the lacr/o, the tetr/o and arac/i1-i2 regulatory elements," *Nucleic acids research*, vol. 25, no. 6, pp. 1203–1210, 1997.

[62] J. Gonçalves 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.

[63] V. Chetty and S. Warnick, "Network semantics of dynamical systems," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 1557–1562.

[64] E. Yeung, J. Gonçalves, H. Sandberg, and S. Warnick, "Mathematical relationships between representations of structure in linear interconnected dynamical systems," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 4348–4353.

[65] C. A. Johnson, N. Woodbury, and S. Warnick, "Graph theoretic foundations of cyclic and acyclic linear dynamic networks," 2020.

[66] J. Adebayo, T. Southwick, V. Chetty, E. Yeung, Y. Yuan, J. Goncalves, J. Grose, J. Prince, G.-B. Stan, and S. Warnick, "Dynamical structure function identifiability conditions enabling signal structure reconstruction," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4635–4641.

[67] E. Yeung, J. Kim, Y. Yuan, J. Gonçalves, and R. Murray, "Quantifying crosstalk in biochemical systems," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 5528–5535.

[68] N. A. Cookson, W. H. Mather, T. Danino, O. Mondragón-Palomino, R. J. Williams, L. S. Tsimring, and J. Hasty, "Queueing up for enzymatic processing: correlated signaling through coupled degradation," *Molecular systems biology*, vol. 7, no. 1, p. 561, 2011.

[69] F. Ceroni, A. Boo, S. Furini, T. E. Gorochowski, O. Borkowski, Y. N. Ladak, A. R. Awan, C. Gilbert, G.-B. Stan, and T. Ellis, "Burden-driven feedback control of gene expression," *Nature methods*, vol. 15, no. 5, pp. 387–393, 2018.

[70] C. Thomaseth, D. Fey, T. Santra, O. S. Rukhlenko, N. E. Radde, and B. N. Kholodenko, "Impact of measurement noise, experimental design, and estimation methods on modular response analysis based network reconstruction," *Scientific reports*, vol. 8, no. 1, pp. 1–14, 2018.

[71] A. Gyorgy, J. I. Jiménez, J. Yazbek, H.-H. Huang, H. Chung, R. Weiss, and D. Del Vecchio, "Isocost lines describe the cellular economy of genetic circuits," *Biophysical Journal*, vol. 109, no. 3, pp. 639–646, 2015.

[72] J. Kim and E. Winfree, "Synthetic in vitro transcriptional oscillators," *Molecular systems biology*, vol. 7, no. 1, p. 465, 2011.

[73] L. Ljung, *System Identification–Theory for the User*. Prentice Hall, 1999.

[74] V. Noireaux, R. Bar-Ziv, and A. Libchaber, "Principles of cell-free genetic circuit assembly," *Proceedings of the National Academy of Sciences*, vol. 100, no. 22, pp. 12 672–12 677, 2003.

[75] D. Del Vecchio, A. J. Ninfa, and E. D. Sontag, "Modular cell biology: retroactivity and insulation," *Molecular systems biology*, vol. 4, no. 1, p. 161, 2008.

[76] V. Hsiao, Y. Hori, P. W. Rothemund, and R. M. Murray, "A population-based temporal logic gate for timing and recording chemical events," *Molecular systems biology*, vol. 12, no. 5, p. 869, 2016.

[77] A. E. Friedland, T. K. Lu, X. Wang, D. Shi, G. Church, and J. J. Collins, "Synthetic gene networks that count," *science*, vol. 324, no. 5931, pp. 1199–1202, 2009.

[78] L. B. Andrews, A. A. Nielsen, and C. A. Voigt, "Cellular checkpoint control using programmable sequential logic," *Science*, vol. 361, no. 6408, 2018.

[79] C. Engler, R. Gruetzner, R. Kandzia, and S. Marillonnet, "Golden gate shuffling: a one-pot dna shuffling method based on type iis restriction enzymes," *PloS one*, vol. 4, no. 5, p. e5553, 2009.

[80] D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie *et al.*, "Creation of a bacterial cell controlled by a chemically synthesized genome," *science*, vol. 329, no. 5987, pp. 52–56, 2010.

[81] C. Engler, R. Kandzia, and S. Marillonnet, "A one pot, one step, precision cloning method with high throughput capability," *PloS one*, vol. 3, no. 11, p. e3647, 2008.

[82] E. Weber, C. Engler, R. Gruetzner, S. Werner, and S. Marillonnet, "A modular cloning system for standardized assembly of multigene constructs," *PloS one*, vol. 6, no. 2, p. e16765, 2011.

[83] V. K. Mutalik, J. C. Guimaraes, G. Cambray, C. Lam, M. J. Christoffersen, Q.-A. Mai, A. B. Tran, M. Paull, J. D. Keasling, A. P. Arkin *et al.*, "Precise and reliable gene expression via standard transcription and translation initiation elements," *Nature methods*, vol. 10, no. 4, pp. 354–360, 2013.

[84] Y.-J. Chen, P. Liu, A. A. Nielsen, J. A. Brophy, K. Clancy, T. Peterson, and C. A. Voigt, "Characterization of 582 natural and synthetic terminators and quantification of their design constraints," *Nature Methods*, vol. 10, no. 7, pp. 659–664, 2013.

[85] M. A. Savageau, "Design principles for elementary gene circuits: Elements, methods, and examples," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 11, no. 1, pp. 142–159, 2001.

[86] R. Milo, P. Jorgensen, U. Moran, G. Weber, and M. Springer, "Bionumbers: the database of key numbers in molecular and cell biology," *Nucleic acids research*, vol. 38, no. suppl 1, pp. D750–D753, 2010.

[87] Y. Yuan, G.-B. Stan, S. Warnick, and J. Goncalves, "Robust dynamical network structure reconstruction," *Automatica*, vol. 47, no. 6, pp. 1230–1235, 2011.

[88] J. B. Andersen, C. Sternberg, L. K. Poulsen, S. P. Bjørn, M. Givskov, and S. Molin, "New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria," *Applied and environmental microbiology*, vol. 64, no. 6, pp. 2240–2246, 1998.

X. Supplementary Information

Experimental Methods for Circuit Preparation, Assembly, and Testing

A. The Repressilator Genelet Circuit

The DNA sequences for the T31, T12, T23 switch were obtained as a gift from the Winfree lab, mirroring the design identically of the repressilator genelet circuit used in [72]. Oligonucleotides were ordered with functionalized fluorophores or quenchers, corresponding to the original design of the genetic repressilator. DNA sequences were suspended in Tris-EDTA buffer for primary stock storage, while all genelet switches T12, T31, T23 added at concentrations of 75 nM, 75 nM, and 60 nM, respectively to match previous tuning experiments to balance the repressilator, with 7.5 mM working concentration of mono-NTP solution, 24 mM MgCl2, and 1x T7 expression system buffer.

DNA analogues of RNA inhibitors were added to sequester DNA activator signal from the switches as an effective step input perturbation to each node. The switches produced a RNA signal that was designed to interfere with formation of a complete promoter region of the next downstream switch in the repressilator circuit. Adding DNA served as a step perturbation to the corresponding switch. Each DNA moiety added thus had the effect of an activator. Activator DNA molecules A1, A2, and A3, each containing Iowa Black quencher were added at 75 nM, 80 nM, and 75 nM working concentration at 20 minutes from the onset of the reaction, to determine the maximum range of quenching. At 58 minutes, we added 0.7 μL of pyrophosphatase, 3 μL of T7 RNA Polymerase and 2.2 μL of RNase H to achieve identical working concentrations as those described in [72].

B. The Transcriptional Event Detector Circuit

The transcriptional event detector circuit, as illustrated in Figure 7 in the main text, is composed of four distinct gene expression cassettes that define the regulatory logic of the circuit and four distinct gene expression cassettes that generate the fluorescent reporter elements of the circuit. Each gene cassette defines a transcriptional unit, with a promoter element, an RBS, a coding sequence, and a terminator sequence. Each gene cassette was cloned using a 5 part Golden Gate assembly, with a type II BsaI restriction enzyme and overhang sequences from [81], [82]. Each assembled gene cassette was cloned in JM109 *E. coli* cloning strains and sequence verified at Eurofins Genomic, by Sanger sequencing. Assembled plasmids were engineered to enable a second stage Golden Gate assembly, using the BbsI Type II restriction enzyme, and assembled to either 1) form a master regulatory logic plasmid (pEY15K), comprised of four distinct gene expression cassettes driving transcription factor or allosteric response or 2) form a master reporter plasmid comprised of four distinct reporter elements (pEY14C). Both Stage 2 assembled regulatory logic and reporter plasmids were sequence verified using Sanger sequencing (Eurofin Genomics) and transformed into MG1655ΔLacI (a gift from the Collins laboratory). The sequences for all individual plasmids and the circuit plasmids are listed in Table I.

Sequences of Genetic Circuit Components

The sequences for all genetic components and circuits for the event detector circuit are listed in Table I. All genelet repressilator sequences are identical to the sequences used and listed in [72]. All ribosome binding site (RBS) sequences were derived from the bicistronic design (BCD) ribosome binding site library [83], while all terminator sequences were drawn from the synthetic terminator library characterized in [84].

Sequence ID	Sequence Description	DNA Sequence
pAra-Lac	Hybrid Promoter	CATAGCATTTTTATCCATAAGATTAGCGGATCCTAAGCTTTACAA TTGTGAGCGCTCACAATTATGATAGATTCAATTGTGAGCGGATA ACAATTTCACACA
BCD2	RBS	GGGCCCAAGTTCACTTAAAAAGGAGATCAACAATGAAAGCAATT TTCGTACTGAAACATCTTAATCATGCAGGGGAGGGTTTCTAATG
TetR	Transcription Factor	ATGTCTAGATTAGATAAAAGTAAAGTGATTAACAGCGCATTAGAG CTGCTTAATGAGGTCGGAATCGAAGGTTTAACAACCCGTAAACT CGCCCAGAAGCTAGGTGTAGAGCAGCCTACATTGTATTGGCATG TAAAAAATAAGCGGGCTTTGCTCGACGCCTTAGCCATTGAGATGT TAGATAGGCACCATACTCACTTTTGCCCTTTAGAAGGGGAAAGCT GGCAAGATTTTTTACGTAATAACGCTAAAAGTTTTAGATGTGCTTTA CTAAGTCATCGCGATGGAGCAAAAGTACATTTAGGTACACGGCCTA CAGAAAAACAGTATGAAACTCTCGAAAATCAATTAGCCTTTTTATGC CAACAAGGTTTTTCACTAGAGAATGCATTATATGCACTCAGCGCTGT GGGGCATTTTACTTTAGGTTGCGTATTGGAAGATCAAGAGCATCAAG TCGCTAAAGAAGAAAGGGAAACACCTACTACTGATAGTATGCCGCCA TTATTACGACAAGCTATCGAATTATTTGATCACCAAGGTGCAGAGCCA GCCTTCTTATTCGGCCTTGAATTGATCATATGCGGATTAGAAAAACAA CTTAAATGTGAAAGTGGGTCTGCAGCAAACGACGAAAACTACGCTTT AGCAGCTTAA
ECK120033736	Terminator	AACGCATGAGAAAGCCCCCGGAAGATCACCTTCCGGGGGCTTTTTT ATTGCGC
BCD9	RBS	GGGCCCAAGTTCACTTAAAAAGGAGATCAACAATGAAAGCAATTTTC GTACTGAAACATCTTAATCATGCAGAGGAGTCTTTCT
AraC	Transcription Factor	ATGCAATATGGACAATTGGTTTCTTCTCTGAATGGCGGGAGTATGAA AAGTATGGCTGAAGCGCAAAATGATCCCCTGCTGCCGGGATACTCG TTTAATGCCCATCTGGTGGCGGGTTTAACGCCGATTGAGGCCAACG GTTATCTCGATTTTTTTATCGACCGACCGCTGGGAATGAAAGGTTATA TTCTCAATCTCACCATTCGCGGTCAGGGGGTGGTGAAAAATCAGGG ACGAGAATTTGTTTGCCGACCGGGTGATATTTTGCTGTTCCCGCCAG GAGAGATTCATCACTACGGTCGTCATCCGGAGGCTCGCGAATGGTAT CACCAGTGGGTTTACTTTCGTCCGCGCGCCTACTGGCATGAATGGCT TAACTGGCCGTCAATATTTGCCAATACGGGGTTCTTTCGCCCGGATGA AGCGCACCAGCCGCATTTCAGCGACCTGTTTGGGCAAATCATTAACG CCGGGCAAGGGGAAGGGCGCTATTCGGAGCTGCTGGCGATAAATCT GCTTGAGCAATTGTTACTGCGGCGCATGGAAGCGATTAACGAGTCGC

		TCCATCCACCGATGGATAATCGGGTACGCGAGGCTTGTCAGTACATCA GCGATCACCTGGCAGACAGCAATTTTGATATCGCCAGCGTCGCACAGC ATGTTTGCTTGTCGCCGTCGCGTCTGTCACATCTTTTCCGCCAGCAGTT AGGGATTAGCGTCTTAAGCTGGCGCGAGGACCAACGTATCAGCCAGGC GAAGCTGCTTTTGAGCACCACCCGGATGCCTATCGCCACCGTCGGTCG CAATGTTGGTTTTGACGATCAACTCTATTTCTCGCGGGTATTTAAAAAATG CACCGGGGCCAGCCCGAGCGAGTTCCGTGCCGGTTGTGAAGAAAAAGT GAATGATGTAGCCGTCAAGTTGTCATAA
ECK120029600	Terminator	TTCAGCCAAAAAACTTAAGACCGCCGGTCTTGTCCACTACCTTGCAGTA ATGCGGTGGACAGGATCGGCGGTTTT CTTTTCTCTTCTCAA
pLas-Tet	Hybrid Promoter	TTCTTCGAGCCTAGCAAGGGTCCGGGTTCACCGAAATCTA TCTCATTTGCTAGTTATAAAATTATGAAATTTGCGTAAATTCC CTATCAGTGATAGAGATTCAGAAGC
BCD10	RBS	GGGCCCAAGTTCACTTAAAAAGGAGATCAACAATGAAAGCA ATTTTCGTACTGAAACATCTTAATCATGCGGAGGATCGTTTCTA
LacI	Transcription Factor	ATGAAACCAGTAACGTTATACGATGTCGCAGAGTATGCCGGTG TCTCTTATCAGACCGTTTCCCGCGTGGTGAACCAGGCCAGCC ACGTTTCTGCGAAAACGCGGGAAAAAGTGGAAGCGGCGATG GCGGAGCTGAATTACATTCCCAACCGCGTGGCACAACAACTG GCGGGCAAACAGTCGTTGCTGATTGGCGTTGCCACCTCCAGT CTGGCCCTGCACGCGCCGTCGCAAATTGTCGCGGCGATTAAA TCTCGCGCCGATCAACTGGGTGCCAGCGTGGTGGTGTCGATG GTAGAACGAAGCGGCGTCGAAGCCTGTAAAGCGGCGGTGCAC AATCTTCTCGCGCAACGCGTCAGTGGGCTGATCATTAACTATCC GCTGGATGACCAGGATGCCATTGCTGTGGAAGCTGCCTGCACT AATGTTCCGGCGTTATTTCTTGATGTCTCTGACCAGACACCCATC AACAGTATTATTTTCTCCCATGAGGACGGTACGCGACTGGGCGT GGAGCATCTGGTCGCATTGGGTCACCAGCAAATCGCGCTGTTAG CGGGCCCATTAAGTTCTGTCTCGGCGCGTCTGCGTCTGGCTGGC TGGCATAAATATCTCACTCGCAATCAAATTCAGCCGATAGCGGAAC GGGAAGGCGACTGGAGTGCCATGTCCGGTTTTCAACAAACCATG CAAATGCTGAATGAGGGCATCGTTCCCACTGCGATGCTGGTTGC CAACGATCAGATGGCGCTGGGCGCAATGCGCGCCATTACCGAGT CCGGGCTGCGCGTTGGTGCGGATATCTCGGTAGTGGGATACGAC GATACCGAGGACAGCTCATGTTATATCCCGCCGTTAACCACCATCA AACAGGATTTTCGCCTGCTGGGGCAAACCAGCGTGGACCGCTTG CTGCAACTCTCTCAGGGCCAGGCGGTGAAGGGCAATCAGCTGTTG CCCGTCTCACTGGTGAAAAGAAAAACCACCCTGGCGCCCAATACGC AAACCGCCTCTCCCCGCGCGTTGGCCGATTCATTAATGCAGCTGGC ACGACAGGTTTCCCGACTGGAAAGCGGGCAGGCAGCAAACGACGA AAACTACGCTTTAGCAGCTTGA
ECK120015440	Terminator	TCCGGCAATTAAAAAAGCGGCTAACCACGCCGCTTTTTTTACGTCTGCA
pLas	LasR promoter	GCATTGCTGTTCTTGATGGCTAGCTCAGTCCTAGGTACAATGCAAGC
BCD1	RBS	GGGCCCAAGTTCACTTAAAAAGGAGATCAACAATGAAAGCAATTTTCG TACTGAAACATCTTAATCATGCACAGGAGACTTTCTAATG
LasR	Transcription Factor	ATGGCCTTGGTTGACGGTTTTCTTGAGCTGGAACGCTCA AGTGGAAAATTGGAGTGGAGCGCCATCCTCCAGAAGATG GCGAGCGACCTTGGATTCTCGAAGATCCTGTTCGGCCTG TTGCCTAAGGACAGCCAGGACTACGAGAACGCCTTCATC GTCGGCAACTACCCGGCCGCCTGGCGCGAGCATTACGA CCGGGCTGGCTACGCGCGGGTCGACCCGACGGTCAGTC ACTGTACCCAGAGCGTACTGCCGATTTTCTGGGAACCGTC CATCTACCAGACGCGAAAGCAGCACGAGTTCTTCGAGGAA GCCTCGGCCGCCGGCCTGGTGTATGGGCTGACCATGCCG CTGCATGGTGCTCGCGGCGAACTCGGCGCGCTGAGCCTC AGCGTGGAAGCGGAAAACCGGGCCGAGGCCAACCGTTTC ATAGAGTCGGTCCTGCCGACCCTGTGGATGCTCAAGGACT ACGCACTGCAAAGCGGTGCCGGACTGGCCTTCGAACATC CGGTCAGCAAACCGGTGGTTCTGACCAGCCGGGAGAAGG AAGTGTTGCAGTGGTGCGCCATCGGCAAGACCAGTTGGGA GATATCGGTTATCTGCAACTGCTCGGAAGCCAATGTGAACTT CCATATGGGAAATATTCGGCGGAAGTTCGGTGTGACCTCCC GCCGCGTAGCGGCCATTATGGCCGTTAATTTGGGTCTTATT ACTCTCTAATAA
ECK120010799	Terminator	GTTATGAGTCAGGAAAAAAGGCGACAGAGTAATCTGTCGCC TTTTTTCTTTGCTTGCTTT
CFP	CDS	ATGAGTAAAGGAGAAGAACTTTTCACTGGAGTTGTCCCAATTC TTGTTGAATTAGATGGTGATGTTAATGGGCACAAATTTTCTGTC AGTGGAGAGGGTGAAGGTGATGCAACATACGGAAAACTTACC CTTAAATTTATTTGCACTACTGGAAAACTACCTGTTCCATGGCC AACACTTGTCACTACTTTGACTTGGGGTGTTCAATGCTTTGCTA GATACCCAGATCATATGAAACAGCATGACTTTTTCAAGAGTGCC TGCCCGAAGGTTATGTACAGGAAAGAACTATATTTTTCAAAGAT GACGGGAACTACAAGACACGTGCTGAAGTCAAGTTTGAAGGT GATACCCTTGTTAATAGAATCGAGTTAAAAGGTATTGATTTTAAA GAAGATGGAAACATTCTTGGACACAAATTGGAATACAACGCTAT TTCAGATAATGTATACATCACTGCAGACAAACAAAAGAATGGAAT CAAAGCTAATTTCAAAATTAGACACAACATTGAAGATGGAAGCGT TCAACTAGCAGACCATTATCAACAAAATACTCCAATTGGCGATGGC CCTGTCCTTTTACCAGACAACCATTACCTGTCCACACAATCTGCCC TTTCGAAAGATCCCAACGAAAAGAGAGATCACATGGTCCTTCTTGAG TTTGTAACAGCTGCTGGGATTACACTAGGCATGGATGAACTATACAAA
Citrine	CDS	ATGTCTAAAGGTGAAGAATTATTCACTGGTGTTGTCCCAATTTTGGTT GAATTAGATGGTGATGTTAATGGTCACAAATTTTCTGTCTCCGGTGAA GGTGAAGGTGATGCTACTTACGGTAAATTGACCTTAAAATTTATTTGTA CTACTGGTAAATTGCCAGTTCCATGGCCAACCTTAGTCACTACTTTAG GTTATGGTTTGATGTGTTTTGCTAGATACCCAGATCATATGAAACAACA TGACTTTTTCAAGTCTGCCATGCCAGAAGGTTATGTTCAAGAAAGAAC TATTTTTTTCAAAGATGACGGTAACTACAAGACCAGAGCTGAAGTCAAG TTTGAAGGTGATACCTTAGTTAATAGAATCGAATTAAAAGGTATTGATTTTA AAGAAGATGGTAACATTTTAGGTCACAAATTGGAATACAACTATAACTCTC ACAATGTTTACATCATGGCTGACAAACAAAAGAATGGTATCAAAGTTAACT TCAAAATTAGACACAACATTGAAGATGGTTCTGTTCAATTAGCTGACCATT ATCAACAAAATACTCCAATTGGTGATGGTCCAGTCTTGTTACCAGACAAC CATTACTTATCCTATCAATCTAGATTATCCAAAGATCCAAACGAAAAGAGAG ATCACATGGTCTTGTTAGAATTTGTTACTGCTGCTGGTATTACCCATGGTAT GGATGAATTGTACAAA
mRFP	CDS	ATGGCTTCCTCCGAAGATGTTATCAAAGAGTTCATGCGTTTCAAAGTTCGT ATGGAAGGTTCCGTTAACGGTCACGAGTTCGAAATCGAAGGTGAAGGTG AAGGTCGTCCGTACGAAGGTACCCAGACCGCTAAACTGAAAGTTACCAA AGGTGGTCCGCTGCCGTTCGCTTGGGACATCCTGTCCCCGCAGTTCCA GTACGGTTCCAAAGCTTACGTTAAACACCCGGCTGACATCCCGGACTAC CTGAAACTGTCCTTCCCGGAAGGTTTCAAATGGGAACGTGTTATGAACT TCGAGGACGGTGGTGTTGTTACCGTTACCCAGGACTCCTCCCTGCAAG ACGGTGAGTTCATCTACAAAGTTAAACTGCGTGGTACCAACTTCCCGTC CGACGGTCCGGTTATGCAGAAAAAAACCATGGGTTGGGAAGCTTCCAC CGAACGTATGTACCCGGAAGATGGTGCTCTGAAAGGTGAAATCAAAATG CGTCTGAAACTGAAAGACGGTGGTCACTACGACGCTGAAGTTAAAACC ACCTACATGGCTAAAAAACCGGTTCAGCTGCCGGGTGCTTACAAAACCG ACATCAAACTGGACATCACCTCCCACAACGAGGACTACACCATCGTTGA ACAGTACGAACGTGCTGAAGGTCGTCACTCCACCGGTGCTTAA

TABLE I: Table of genetic sequences for all parts used to make the transcriptional event detector circuit

XI. Data Accessibility

All data files and network reconstruction code can be obtained from the GitHub repository `https://github.com/YeungRepo/NetworkRecon`.

XII. Quantifying Crosstalk in Biochemical Reaction Networks

A common way that crosstalk arises in biochemical reaction networks is when species compete for commonly shared enzymes. When this occurs, the sequestration of an enzyme by one competing species makes the enzyme less accessible to other competing species. For example, when two mRNA are competing for a single ribosome, the binding of one mRNA to the ribosome during translation makes it less accessible to other mRNA. At the core of any such crosstalk is a sudden increase in the dependency of one biochemical state on another. Though enzyme loading may be a common source of crosstalk, such interactions can be modeled at a higher level of abstraction, namely how the dynamics of a given state are affected by the concentration fluctuations of other states.

Nearly every synthetic gene network implements causal dependencies among states. Often, these "designed" interactions take the form of transcription factor binding, sense-anti-sense mRNA regulation, and sequestration events. In practice, every physical system exhibits trajectories that are a mixture of the consequences of both interaction types: designed and crosstalk interactions. Throughout the course of this paper, we will denote the physical system of interest in our models as

$$\begin{aligned} \dot{y} &= f_y^c(y, x_h, u), \quad y(0) = y_0 \\ \dot{x}_h &= f_{x_h}^c(y, x_h, u), \quad x_h(0) = x_{h,0}, \\ y &= \begin{bmatrix} \mathbf{I}_{p\times p} & 0 \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix} \end{aligned} \tag{15}$$

To quantify crosstalk in such systems, we can compare the dynamics of system (15) against the dynamics of a reference or alternative system that is free of crosstalk. Such a reference system will still retain the *desired* interaction dynamics and reflects the idealized model often used to design a synthetic gene network, e.g. the feed-forward loop model in Example II-A1. Moreover, it can represent the desired behavior of the system in a regime where the magnitude of crosstalk effects are supposed to be minimal or engineered in such a way that they are suppressed [7]. We write the reference system as

$$\begin{aligned} \dot{y} &= f_y^a(y, x_h, u), \quad y(0) = y_0 \\ \dot{x}_h &= f_{x_h}^a(y, x_h, u), \quad x_h(0) = x_{h,0}, \\ y &= \begin{bmatrix} \mathbf{I}_{p\times p} & 0 \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix} \end{aligned} \tag{16}$$

Remark 1: For the comparison between the alternative and crosstalk system to be fair, it is important that (16) satisfies *internal equivalence* [85]. Specifically, we will suppose that any parameters or dynamics unassociated with crosstalk, e.g. interaction dynamics, catalytic reactions, or anabolic reactions with no loading effects, are held fixed. Thus, as we compare the behavior of both systems, any differences in the hidden state x_h or output y dynamics are purely due to effects of crosstalk.

With the definition of an alternative system in place, it becomes possible to reason about the size of crosstalk, by comparing the dynamics of both systems. In particular, we can develop a rigorous notion for describing the amount of crosstalk arising from the difference of trajectories in both systems.

Definition 1 (Crosstalk Trajectory): Consider two systems, a crosstalk system and an alternative or reference system, initialized from the same initial condition $x(0)$. For each initial condition $x(0) = (y(0), x_h(0)) \in \mathbb{R}^n$ and input trajectory $u(t)$ we define the *crosstalk trajectory* $\zeta(t)$ as

$$\zeta(t) = x^a(t) - x^c(t)$$

The crosstalk trajectory is a time-evolving vector that describes the deviation of the physical system (subject to crosstalk) from the reference system's trajectory. With this notion of crosstalk, we can also make precise the concept of crosstalk between states. We note that in writing the following quantity of interest $\frac{\partial}{\partial x_j}\zeta_i$, it is with a slight abuse of notation, since $\zeta_i(x^a(t), x^c(t))$. Mathematically, we are computing the jth partial derivative of each term in $\zeta_i = x_i^a(t) - x_i^c(t)$. Thus, to be clear, when we write $\frac{\partial}{\partial x_j}\zeta_i$, it will be implicit that we mean $\frac{\partial}{\partial x_j^a}x_i^a(t) - \frac{\partial}{\partial x_j^c}x_i^c(t)$.

Definition 2 (Directed Crosstalk): Given an initial condition of $(x(0), y(0))$ and input trajectory $u(t)$ we say that a chemical species x_j exerts a crosstalk effect on chemical species x_i if the ith component of the crosstalk trajectory $\zeta(t)$ has nonzero partial derivative

$$\frac{\partial}{\partial x_j}\zeta_i(t) \neq 0.$$

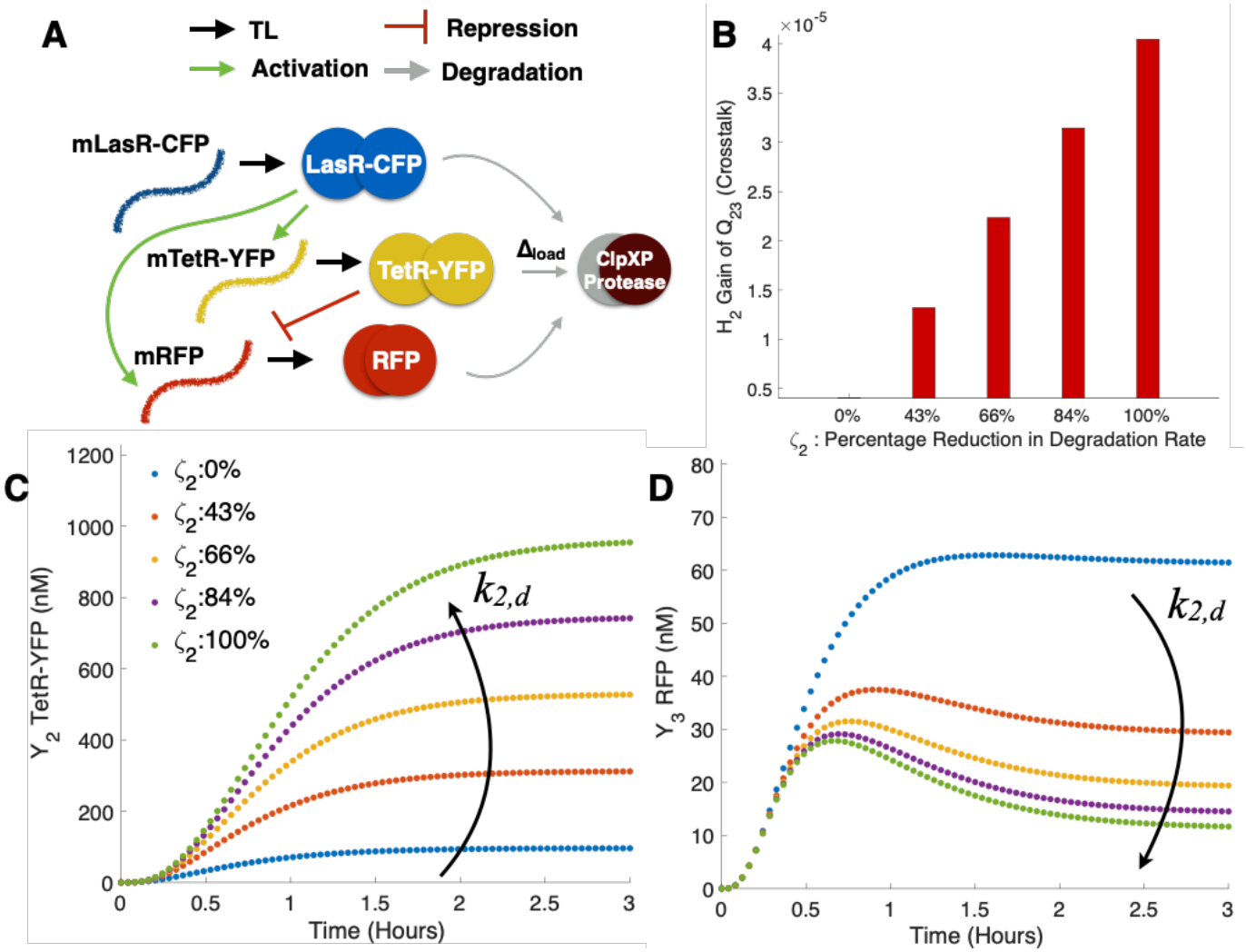

Fig. 11. **Dynamical structure functions quantify biomolecular crosstalk:(A)** A schematic illustrating the design of this simulation example. The crosstalk and reference model of the incoherent feedforward loop from Examples II-A1 and II-A2 are simulated accordingly to satisfy internal equivalence, for varying values of $k_{2,d}$. Standard parameters from the literature [86] were used to generate the simulation. As the size of the load Δ_{load} increases, the ability of the IFFL to respond with a pulse decreases.**(B)** The $\mathcal{H}_2$ gain of $Q^c_{23}(s)$ is plotted as a function of ζ. Notice that $Q^c_{23}(s)$ is a pure crosstalk term, since $Q^a_{23}(s) \equiv 0$. As the effective crosstalk in ζ_2 increases, $Q^c_{23}(s)$ mirrors that increase, as shown in Proposition 3. **(C-D)** Time lapse responses of the incoherent feedforward loop: for each value of $k_{2,d}$ the value of ζ_2 at $t = 3$ hours is calculated and used to label curves (as percentage of maximum load). Notice the monotonic relationship between $k_{2,d}$, ζ and the output responses of Y_2 and Y_3 (negatively monotonic).

for some initial condition of $(x(0), y(0))$ and input trajectory $u(t)$. In general, we will refer to $\frac{\partial}{\partial x_j}\zeta_i(t)$ as the *crosstalk sensitivity* of x_i to x_j.

Notice that the mathematical definition of crosstalk sensitivity $\frac{\partial}{\partial x_j}\zeta_i(t)$ depends on the initial condition $x_0(t)$ and the input $u(t)$. This dependency is consistent with the parametric sensitivity of biological function. Many genetic circuits in bacteria behave acceptably in one initial condition and for one input condition, e.g., in log-phase with an attenuated amount of a small molecule or sugar compound, but exhibit significantly different behavior when input concentrations are increased by an order of magnitude or subject to an alternate preparation method prior to the experiment. The latter imposes a state history that defines a distinct initial condition, which can drive a biological network to a highly coupled or decoupled state.

Example 1: Consider two mRNA species m_1 and m_2 competing for the same degradation enzyme D in a physical system. For simplicity of exposition, suppose their production dynamics do not depend on each other and can be modeled as $P_1(t)$ and $P_2(t)$ respectively. The crosstalk system is given as

$$\dot{m}_1 = P_1(t) - \frac{D_0 m_1/k_{M,1}}{1 + m_1/k_{M,1} + m_2/k_{M,2}}$$
$$\dot{m}_2 = P_2(t) - \frac{D_0 m_2/k_{M,2}}{1 + m_1/k_{M,1} + m_2/k_{M,2}}$$

while the reference system is given as

$$\dot{m}_1 = P_1(t) - \frac{D_0 m_1/k_{M,1}}{1 + m_1/k_{M,1}}$$
$$\dot{m}_2 = P_2(t) - \frac{D_0 m_2/k_{M,2}}{1 + m_2/k_{M,2}}.$$

In both systems, we have supposed that time has been rescaled so that the customary parameter k_{cat} for degradation is unity. The crosstalk sensitivity of m_1 and m_2 (with respect to each other) are given as

$$\frac{\partial \zeta_1}{\partial m_2} = \frac{\partial}{\partial m_2} \int_0^t \frac{-D_0\, m_1/k_{M,1}\ m_2/k_{M,2}}{(1 + m_1/k_{M,1})(1 + m_1/k_{M,1} + m_2/k_{M,2})}$$
$$\frac{\partial \zeta_2}{\partial m_1} = \frac{\partial}{\partial m_1} \int_0^t \frac{-D_0 m_1/k_{M,1}\ m_2/k_{M,2}}{(1 + m_2/k_{M,2})(1 + m_1/k_{M,1} + m_2/k_{M,2})}$$

respectively. The crosstalk sensitivity between m_1 and m_2 is nonzero whenever m_1 or m_2 have non-zero initial condition.

Remark 2: In synthetic biocircuit design, two chemical species x_i and x_j are often declared orthogonal when there is no designed interaction between them. Mathematically, in the crosstalk free system, this corresponds to

$$\frac{\partial}{\partial x_j} x_i^a(t) \equiv 0$$

for all $x(0)$ and $u(t)$. In such a situation, $\zeta(x_i, x_j) \neq 0$ if and only if

$$\frac{\partial}{\partial x_j} x_i^c(t) = \int_0^t f_i^c(y, x_h, u) d\tau \neq 0.$$

This condition is interesting in experimental settings since a computational estimate of $\frac{\partial}{\partial x_j} \int_0^t f_i^c(t)$ from perturbation experiments coincides with a direct estimate of the sensitivity of the crosstalk $\frac{\partial}{\partial x_j} \zeta_i$. More specifically, when x_i and x_j are measured outputs of the system, we will show in the sequel that quantifying $||Q_{i,j}^c(s)||$ is directly related to an estimate of the crosstalk sensitivity $\frac{\partial}{\partial x_j} \zeta_i(t)$ near the equilibrium point x_e^c.

Remark 3: In general, estimating the crosstalk sensitivity for the nonlinear systems (15) and (16) can be challenging if either x_i and x_j are not measured directly. Firstly, if experimental data is available, it will often consist of data for the measured species y in the crosstalk-system, but not the reference system. Second, if only one of the species x_i (or none) is available for measurement, even if perturbation of x_j is possible, a nonlinear observer is required to estimate the trajectory of $x_j(t)$. Unless the parameters of $f_i(x, u)$ are known *a priori* (which is generally not the case), this then also requires system identification of the parameters of $f_c(x, u)$ and $f_a(x, u)$ which often results in a non-convex optimization problem.

Thus, our goal is to estimate the observed crosstalk between measured species Y_i and Y_j. This crosstalk estimate will invariably include the dynamics of unmeasured chemical species (such as ATP, RNAP, untagged mRNA and protein species, DNA-protein complexes etc.). From a synthetic biology design standpoint, this is not a disadvantage, since the goal is to design a synthetic gene network with an *abstracted* circuit architecture operating reliably in the context of many unmeasured species. In any genetic circuit, there are always additional biochemical compounds that are unmeasured. Our goal is to validate that a biocircuit (e.g. an IFFL, repressilator, or a novel biocircuit) still manifests the intended network structure even in the presence of unmeasured dynamics.

Theorem 3: Let $\mathcal{L}$ denote the two-sided Laplace operator. Suppose the states x^c and x^a of the systems (15) are (16) are shifted, so that the origin is a locally asymptotically stable equilibrium point and Q^c and Q^a are the respective dynamical structure functions calculated for each linearized system about the origin. Then

$$\frac{\partial \mathcal{L}\left(\zeta_i\right)}{\partial Y_j} = Q_{ij}^a(s) - Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}\left(O(x^2)\right) \tag{17}$$

and in particular, if

$$Q_{ij}^a(s) \equiv 0$$

then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = -Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}\left(O(x^2)\right)$$

and can be estimated from input output data $(Y(s), U(s))$.

Proof 1: First, notice that the Laplace transform of $\mathcal{L}\left(\zeta(t)\right) = \mathcal{L}\left(x^a - x^c\right) \triangleq X^a(s) - X^c(s)$, which can be decomposed

into its measured and unmeasured states

$$\begin{bmatrix} Y^a \\ X_h^a \end{bmatrix}(s) - \begin{bmatrix} Y^c \\ X_h^c \end{bmatrix}(s) = \begin{bmatrix} Q^a Y^a(s) - Q^c Y^c(s) + (P^a - P^c)U(s) + \mathcal{L}\left(O(x^2)\right) \\ X_h^a(s) - X_h^c(s) + \mathcal{L}\left(O(x^2)\right). \end{bmatrix}$$

Examining the ith component equation and taking partials along $Y_j(s)$ yields equation (17).

This result is important, since it tells us when estimating $Q^c(s)$ from experimental data will correspond to estimating crosstalk between measured states in $Y(s)$. Since necessary and sufficient conditions for identifying $Q(s)$ and $P(s)$ have been already characterized [62], this provides conditions for inferring crosstalk from input-output data. For example, a sufficient condition required is that there is an input variable available to excite each measured output of the genetic network attempting to be reconstructed. This allows for the possibility that some biological states are unmeasured and unexcited, but these will be viewed as hidden states that play a role in defining the edge dynamics in $Q^c(s)$.

More generally, even if parameters for $f^a(x,u)(t)$ are unknown, the structure of $Q^a(s)$ can be analytically calculated (using a symbolic algebra package). For every zero entry in $Q^a(s)$ (coinciding with designed orthogonality between measured states), we can then estimate $Q^c(s)$ directly.

In practice, estimation of $Q^c(s)$ is also confounded by noise. In our analysis in this paper, we suppose that a series of filters can be applied to eliminate the noise in the data. This may not be the case for biological systems that have been characterized as inherently stochastic, e.g. single cell gene expression dynamics. In such settings, the estimated dynamical structure $Q^c(s)$ is a mixture of the process noise in the system and the crosstalk. From the standpoint of synthetic biocircuit prototyping, both are undesirable in the ultimate iteration of the biocircuit and thus need to be quantified. In this paper, we will demonstrate our theoretical and computational framework with experimental results derived from *in vitro* systems, where signal-to-noise ratios are high and the only sources of noise are measurement noise and pipetting error. For a theoretical treatment of how to reverse engineer $Q^c(s)$ in the presence of process noise or system perturbation, see [87].

An advantage of using $Q^c(s)$ to estimate the crosstalk is that we can use the $\mathcal{H}_\infty$ norm of $Q^c_{i,j}(s)$ to calculate the worst-case crosstalk magnitude and $\mathcal{H}_2$ of $Q^c_{i,j}(s)$ to calculate the average crosstalk across all frequencies.

1) Quantifying Crosstalk with $Q^c(s)$: Recall the incoherent feedforward loop in subsections II-A1 and II-A2. In particular, comparing $Q^a(s)$ and $Q^c(s)$ we see that $Q^c(s)$ is a full transfer function matrix

$$\begin{pmatrix} 0 & \frac{1.6\cdot 10^{-3}}{s+2.1\cdot 10^{-3}} & \frac{0.041}{s+2.1\cdot 10^{-3}} \\ \frac{\left(1.6\cdot 10^{-3}\right)s+0.048}{s^2+1.5\,s+3.3\cdot 10^{-3}} & 0 & \frac{0.041}{s+2.1\cdot 10^{-3}} \\ \frac{\left(3.8\cdot 10^{-4}\right)s+7.4\cdot 10^{-4}}{s^2+1.6\,s+0.13} & \frac{\left(3.8\cdot 10^{-4}\right)s+4.4\cdot 10^{-4}}{s^2+1.6\,s+0.13} & 0 \end{pmatrix}$$

and $Q^a(s)$ is lower-triangular, reflecting the network structure of the intended IFFL. By examining the upper triangular entries in $Q^c(s)$, we can directly examine the effects of degradation crosstalk. In the lower entries of $Q^c(s)$, these crosstalk effects are confounded with the direct interactions modeled in $Q^a(s)$. Although the gain of the entries in $Q^c(s)$ are small, they nonetheless can have a significant effect on the dynamics of the IFFL.

In Figure 11 we plot the time-lapse response of $y_2(t)$ and $y_3(t)$ for varying parameter values of $k_{2,d}$ in equation (7). The $k_{2,d}$ parameter is a Michaelis constant that determines the effective affinity of substrate x_2 in binding with C_0. As $k_{2,d}$ increases, the affinity of substrate x_2 is diminished, relative to the affinity of x_1 and x_3. Attenuating $k_{2,d}$ can be viewed as similar to swapping out a strong degradation marker for protease degradation with a weaker degradation marker on the species x_2. In the experimental literature, there are multiple degradation markers for proteins that confer varying binding affinities to an associated protease [88]. In our simulation, we consider five potential values for $k_{2,d}$: 500, 1625, 2750, 3875, and $5000\,\mu M$ corresponding to five artificial LVA markers of varying strengths for the protease ClpXP frequently used in *E. coli*.

Notice that as we decrease the affinity of y_2 for ClpXP, this also coincides with an increased ζ_2 crosstalk magnitude. Here, we have computed $\zeta_2 = y_2^c(t) - y_2^a(t)$. We find that $|\zeta_2|$ increases as $k_{2,d}$ increases. In Figure 11B-D, ζ_2 is plotted as a percentage of maximum absolute change across all values of $k_{2,d}$.

We see that the time-lapse response of $y_2(t)$ increases monotonically for all t as the crosstalk $\zeta_2(t)$ increases. This is consistent with biological intuition, since an increase in competition for resource loading (an increase in $k_{2,d}$) results in prolonged lifetimes of each individual y_2 (TetR-YFP) protein. This in turn results in higher repression levels of y_3 in the incoherent feedforward loop. Increased competition for ClpXP from substrates y_3 and y_1 have the effect of damping y_3 dynamics and reinforcing the pulsatile response of the IFFL. The crosstalk in this circuit thus has the effect of effectively strengthening the negative regulation of y_2 on y_3, encouraging the downward transient after $t \approxeq 0.75$ hours. Our network analysis shows we can improve the robustness of an IFFL's pulse by attenuating the relative binding affinity of the repressor to its protease.

In general, crosstalk effects do not necessarily reinforce the feedback architecture of a biocircuit. This underscores the importance of having techniques for quantifying crosstalk in a synthetic gene network and validating that designed interactions are dominant over crosstalk interactions.