Identifying Biochemical Reaction Networks From Heterogeneous Datasets

Wei Pan, Ye Yuan*, Lennart Ljung, Jorge Gonçalves and Guy-Bart Stan

Abstract— In this paper, we propose a new method to identify biochemical reaction networks (i.e. both reactions and kinetic parameters) from heterogeneous datasets. Such datasets can contain (a) data from several replicates of an experiment performed on a biological system; (b) data measured from a biochemical network subjected to different experimental conditions, for example, changes/perturbations in biological inductions, temperature, gene knock-out, gene over-expression, etc. Simultaneous integration of various datasets to perform system identification has the potential to avoid non-identifiability issues typically arising when only single datasets are used.

I. INTRODUCTION

The problem of identifying biological networks from experimental time series data is of fundamental interest in systems and synthetic biology [22]. For example, such information can aid in the design of drugs or of synthetic biology genetic controllers. Tools from system identification [2] can be applied for such purposes. However, most system identification methods produce estimates of model structures based on data coming from a single experiment.

The interest in identification methods able to handle several datasets simultaneously is twofold. Firstly, with the increasing availability of "big data" obtained from sophisticated biological instruments, e.g. large 'omics' datasets, attention has turned to the efficient and effective integration of these data and to the maximum extraction of information from them. Such datasets can contain (a) data from replicates of an experiment performed on a biological system of interest under identical experimental conditions; (b) data measured from a biochemical network subjected to different experimental conditions, for example, different biological inducers, temperature, stress, optical input, gene knock-out and over-expression, etc. The challenges for simultaneously

considering heterogeneous datasets during system identification are: (a) the system itself is unknown, i.e. neither the structure nor the corresponding parameters are known; (b) it is unclear how heterogeneous datasets collected under different experimental conditions influence the "quality" of the identified system.

Secondly, in control or synthetic biology applications the systems to be controlled typically need to be modelled first. Highly detailed or complex models are typically difficult to handle using rigorous control design methods. Therefore, one typically prefers to use simple or sparse models that capture at best the dynamics expressed in the collected data. The identification and use of simple or sparse models inevitably introduces model class uncertainties and parameter uncertainties [3], [4]. To assess these uncertainties replicates of multiple experiments is typically necessary.

Our approach is based on the concept of sparse Bayesian learning [5], [6] and on the definition of a unified optimisation problem allowing the consideration of different parameter values for different experimental conditions, and whose solution is a model consistent with all datasets available for identification. The ability to consider various datasets simultaneously can potentially avoid non-identifiability issues arising when a single dataset is used [7]. Furthermore, by comparing the identified parameter values associated with different conditions, we can pinpoint the influence specific experimental changes have on system parameters.

The notation in this paper is standard and can be found in SI Section S1.

II. MODEL

We consider dynamical systems described by nonlinear differential/difference equation with additive noise:

$$\delta(x_{nt}) = \mathbf{f}_n(\mathbf{x}_t, \mathbf{u}_t)\mathbf{v}_n + \xi_{nt} \quad i = 1, \ldots, n_{\mathbf{x}}$$
$$= \sum_{s=1}^{N_n} v_{ns} f_{ns}(\mathbf{x}_t, \mathbf{u}_t) + \xi_{nt}, \quad (1)$$

where $\delta(x_{nt}) = \dot{x}_{nt}$ for continuous-time system; $\delta(x_{nt}) = x_{nt}$ or $x_{nt} - x_{n,t-1}$ or some *known* transformation of historical data for discrete-time system; $v_{ns} \in \mathbb{R}$ and $f_{ns}(\mathbf{x}_t, \mathbf{u}_t) : \mathbb{R}^{n_{\mathbf{x}}+n_{\mathbf{u}}} \to \mathbb{R}$ are basis functions that govern the dynamics. To ensure existence and uniqueness of solutions, the functions $f_{ns}(\mathbf{x}_t, \mathbf{u}_t)$ are assumed to be Lipschitz continuous. Note that we do not assume *a priori* knowledge of the form of the nonlinear functions appearing on the right-hand side of the equations in (1), e.g. whether the degradation obeys first-order or enzymatic catalysed dynamics or whether the proteins are repressors or activators.

When data are sampled, we assume the data matrix and first derivative/difference data matrix satisfying (1) can be

obtained as

$$\begin{bmatrix} x_{11} & \dots & x_{n_\mathbf{x}1} \\ \vdots & \ddots & \vdots \\ x_{1M} & \dots & x_{n_\mathbf{x}M} \end{bmatrix} \text{ and } \begin{bmatrix} \delta(x_{11}) & \dots & \delta(x_{n_\mathbf{x}1}) \\ \vdots & \ddots & \vdots \\ \delta(x_{1M}) & \dots & \delta(x_{n_\mathbf{x}M}) \end{bmatrix} \tag{2}$$

respectively.

Based on these defined data matrices, the system in (1) can be written as $\mathbf{y}_n = \boldsymbol{\Psi}_n \mathbf{v}_n + \boldsymbol{\xi}_n$, $n = 1, \dots, n_\mathbf{x}$, where $\mathbf{y}_n \triangleq [\delta(x_{n1}), \dots, \delta(x_{nM})]^\top \in \mathbb{R}^{M \times 1}$, $\mathbf{v}_n \triangleq [v_{n1}, \dots, v_{nN_n}]^\top \in \mathbb{R}^{N_n \times 1}$, $\boldsymbol{\xi}_n \triangleq [\xi_{n1}, \dots, \xi_{nM}]^\top \in \mathbb{R}^{M \times 1}$, and the dictionary matrix $\boldsymbol{\Psi}_n \in \mathbb{R}^{M \times N_n}$ with its j-th column being $[f_{nj}(\mathbf{x}_1, \mathbf{u}_1), \dots, f_{nj}(\mathbf{x}_M, \mathbf{u}_M)]^\top$. The noise or disturbance vector $\boldsymbol{\xi}_n$ is assumed to be Gaussian distributed with zero mean and covariance $\boldsymbol{\Pi} \in \mathbb{R}_+^{M \times M}$ [1]. The identification goal is to estimate $\mathbf{v}_n$ of the linear regression formulation $\mathbf{y}_n = \boldsymbol{\Psi}_n \mathbf{v}_n + \boldsymbol{\xi}_n$, $n = 1, \dots, n_\mathbf{x}$. Two issues need to be raised here. The first one is the selection of basis function $f_{ns}(\cdot, \cdot)$ which is key to the success of identification. Some discussion on this can be found in SI Section S2, especially for biochemical reaction networks. The second one is the estimation of the first derivative data matrix which is not trivial. In SI Section S3 , we provide a method to estimate first derivatives from noisy time-series data.

If a total number of C datasets are collected from C independent experiments, we put a subscript $[c]$ to index the identification problem associated with the specific dataset obtained from experiment $[c]$. In what follows we gather in a matrix $\mathbf{A}_n^{[c]}$ similar to $\boldsymbol{\Psi}_n$ the set of *all* candidate/possible basis functions that we want to consider during the identification. The identification problem is then written as:

$$\mathbf{y}_n^{[c]} = \mathbf{A}_n^{[c]} \mathbf{w}_n^{[c]} + \boldsymbol{\xi}_n^{[c]}, \quad n = 1, \dots, n_\mathbf{x}, \ c = 1, \dots, C. \tag{3}$$

The solution to $\mathbf{w}_n^{[c]}$ to (3) is typically going to be sparse, which is mainly due to the potential introduction of non-relevant and/or non-independent basis functions in $\mathbf{A}_n^{[c]}$.

Since the $n_\mathbf{x}$ linear regression problems in (3) are independent, for simplicity of notation, we omit the subscript n used to index the state variable and simply write:

$$\mathbf{y}^{[c]} = \mathbf{A}^{[c]} \mathbf{w}^{[c]} + \boldsymbol{\xi}^{[c]}, c = 1, \dots, C, \tag{4}$$

in which,

$$\begin{aligned} \mathbf{A}^{[c]} &\triangleq \left[\mathbf{A}_{:,1}^{[c]}, \dots, \mathbf{A}_{:,N}^{[c]} \right] \\ &= \begin{bmatrix} f_1(\mathbf{x}_1^{[c]}) & \dots & f_N(\mathbf{x}_1^{[c]}) \\ \vdots & & \vdots \\ f_1(\mathbf{x}_{M^{[c]}}^{[c]}) & \dots & f_N(\mathbf{x}_{M^{[c]}}^{[c]}) \end{bmatrix} \in \mathbb{R}^{M^{[c]} \times N}, \\ \mathbf{w}^{[c]} &\triangleq \left[w_1^{[c]}, \dots, w_N^{[c]} \right]^\top \in \mathbb{R}^N, \\ \boldsymbol{\xi}^{[c]} &\triangleq \left[\xi_1^{[c]}, \dots, \xi_{M^{[c]}}^{[c]} \right]^\top \in \mathbb{R}^{M^{[c]}}, \end{aligned} \tag{5}$$

where $\mathbf{x}_t^{[c]} = \left[x_{1t}^{[c]}, \dots, x_{n_\mathbf{x}t}^{[c]} \right] \in \mathbb{R}^{n_\mathbf{x}}$ is the state vector at time instant t. It should be noted that N, the number of basis functions or number of columns of the dictionary

[1] Note that the covariance matrix is not necessarily diagonal.

matrix $\mathbf{A}^{[c]} \in \mathbb{R}^{M^{[c]} \times N}$, can be very large. Without loss of generality, we assume $M^{[1]} = \dots = M^{[C]} = M$.

Remark 1: The model class considered in (1) can be enlarged in various ways. First, measurement noise, which is ubiquitous in practice, can be accounted for using the following linear measurement equation:

$$z_t = x_t + \epsilon_t, \tag{6}$$

where the measurement noise ϵ_t is assumed i.i.d. Gaussian. Under this formulation, the noise-contaminated data z_t represents the collected data rather than x_t in (2). Second, the additive stochastic term ξ_t in (1) is often used to model dynamic noise or diffusion. In many practical application, however, it is necessary to account for multiplicative noise instead of additive noise. Multiplicative noise can be accounted for by replacing (1) with $\dot{x}_t = f(\mathbf{x}_t, \mathbf{u}_t)\mathbf{v} + h(\mathbf{x}_t, \mathbf{u}_t)\xi_t$. In SI Section S4, we show how the framework presented here can be modified to encompass these extensions.

III. IDENTIFICATION FROM MULTIPLE DATASETS

To ensure reproducibility, experimentalists repeat their experiments under the same conditions, and the collected data are then called "replicates". Typically, only the average value over these replicates is used for modelling or identification purposes. In this case, however, only the first moment is used and information provided by higher order moments is lost. Moreover, when data originate from different experimental conditions, it is usually very hard to combine the datasets into a single identification problem. This section will address these issues by showing how several datasets can be combined to define a unified optimisation problem whose solution is an identified model consistent with the various datasets available for identification.

To consider heterogeneous datasets in one single formulation, we stack the various equations in (4) (see Eq. (7)). Each stacked equation in Eq. (7) corresponds to a replicate or an experiment performed by changing the experimental conditions on the same system.

In Eq. (7), $\mathbf{A}_i = \text{blkdiag}[\mathbf{A}_{:,i}^{[1]}, \dots, \mathbf{A}_{:,i}^{[C]}]$, and $\mathbf{w}_i = [w_i^{[1]}, \dots, w_i^{[C]}]^\top$, for $i = 1, \dots, N$. Based on the stacked formulation given in Eq. (7) we further define

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} \mathbf{y}^{[1]} \\ \vdots \\ \mathbf{y}^{[C]} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & | & \cdots & | & \mathbf{A}_N \end{bmatrix}, \\ \mathbf{w} &= \begin{bmatrix} \mathbf{w}_1 \\ \hline \vdots \\ \hline \mathbf{w}_N \end{bmatrix}, \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \hline \vdots \\ \hline \boldsymbol{\xi}^{[C]} \end{bmatrix}, \end{aligned} \tag{8}$$

which gives

$$\mathbf{y} = \mathbf{A}\mathbf{w} + \boldsymbol{\xi}. \tag{9}$$

This yields a formulation very similar to that presented previously for a single linear regression problem. However, in the multi-experiment formulation (9), there is now a special block structure for $\mathbf{y}$, $\mathbf{A}$ and $\mathbf{w}$.

Remark 2: When $\mathbf{w}^{[i]}$ is fixed to be $\mathbf{w}$ for all the experiments, i.e. $\mathbf{w}^{[1]} = \dots = \mathbf{w}^{[C]} = \mathbf{w}$, we can formulate the

$$
\begin{bmatrix} \mathbf{y}^{[1]} \\ \vdots \\ \mathbf{y}^{[C]} \end{bmatrix} = \underbrace{\left[\begin{array}{ccc|c|c} \mathbf{A}^{[1]}_{:,1} & \cdots & \mathbf{A}^{[1]}_{:,N} & & \\ & & & \ddots & \\ & & & & \begin{matrix} \mathbf{A}^{[C]}_{:,1} & \cdots & \mathbf{A}^{[C]}_{:,N} \end{matrix} \end{array} \right]}_{C \ \mathbf{Blocks}} \begin{bmatrix} \mathbf{w}^{[1]} \\ \vdots \\ \hline \mathbf{w}^{[C]} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \vdots \\ \hline \boldsymbol{\xi}^{[C]} \end{bmatrix}
$$

$$
= \underbrace{\left[\begin{array}{ccc|c|ccc} \mathbf{A}^{[1]}_{:,1} & & & & \mathbf{A}^{[1]}_{:,N} & & \\ & \ddots & & \ddots & & \ddots & \\ & & \mathbf{A}^{[C]}_{:,1} & & & & \mathbf{A}^{[C]}_{:,N} \end{array} \right]}_{N \ \mathbf{Blocks}} \begin{bmatrix} w^{[1]}_1 \\ \vdots \\ w^{[C]}_1 \\ \hline \vdots \\ \hline w^{[1]}_N \\ \vdots \\ w^{[C]}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \vdots \\ \hline \boldsymbol{\xi}^{[C]} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & | & \cdots & | & \mathbf{A}_N \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \hline \mathbf{w}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \vdots \\ \hline \boldsymbol{\xi}^{[C]} \end{bmatrix}.
$$

$$(7)$$

identification problem as a single linear regression problem by concatenation:

$$
\begin{bmatrix} \mathbf{y}^{[1]} \\ \vdots \\ \mathbf{y}^{[C]} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{[1]} \\ \vdots \\ \mathbf{A}^{[C]} \end{bmatrix} \mathbf{w} + \begin{bmatrix} \boldsymbol{\xi}^{[1]} \\ \vdots \\ \boldsymbol{\xi}^{[C]} \end{bmatrix}. \tag{10}
$$

To incorporate prior knowledge into the identification problem, it is often important to be able to impose constraints on $\mathbf{w}$. In biological systems, positivity of the parameters constituting $\mathbf{w}$ is an example of such constraints. The other motivation for constrained optimisation comes from stability considerations. Typically, the underlying system is known *a priori* to be stable, especially if this system is a biological or physical system. A lot of stability conditions can be formulated as convex optimisation problems, e.g. Lyapunov stability conditions expressed as Linear Matrix Inequalities [8], Gershgorin Theorem for linear systems [9], etc. Only few contributions are available in the literature that address the problem of how to consider *a priori* information on system stability during system identification [10], [11]. To be able to integrate constraints on $\mathbf{w}$ into the problem formulation, we consider the following assumption on $\mathbf{w}$.

Assumption 1: Constraints on the weights $\mathbf{w}$ can be described by a set of convex functions: $H_i^{[I]}(\mathbf{w}) \leq 0, \ \forall i$; $H_j^{[E]}(\mathbf{w}) = 0, \ \forall j$, where the convex functions $H_i^{[I]} : \mathbb{R}^N \to \mathbb{R}$ are used to define inequality constraints, whereas the convex functions $H_j^{[E]} : \mathbb{R}^N \to \mathbb{R}$ are used to define equality constraints.

IV. METHODS

To get an estimate of $\mathbf{w}$ in (9), we use Bayesian modelling to treat all unknowns as stochastic variables with certain probability distributions [13]. For $\mathbf{y} = \mathbf{A}\mathbf{w} + \boldsymbol{\xi}$, it is assumed that the stochastic variables in the vector $\boldsymbol{\xi}$ are Gaussian distributed with *unknown* covariance matrix $\boldsymbol{\Pi}$, i.e., $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Pi})$.

In what follows we consider the following variable substitution for the inverse of unknown covariance matrix or precision matrix: $\mathbf{S} \triangleq \boldsymbol{\Pi}^{-1}$. In such case, the likelihood of the data given $\mathbf{w}$ is

$$
\mathcal{P}(\mathbf{y}|\mathbf{w}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{w}, \boldsymbol{\Pi}) \propto \exp\left[-\frac{1}{2}(\mathbf{A}\mathbf{w} - \mathbf{y})^\top \mathbf{S}(\mathbf{A}\mathbf{w} - \mathbf{y}) \right]. \tag{11}
$$

A. Sparsity Inducing Priors

In Bayesian models, a prior distribution $\mathcal{P}(\mathbf{w})$ can be defined as $\mathcal{P}(\mathbf{w}) = \prod_{i=1}^N \mathcal{P}(\mathbf{w}_i)$ where $\mathcal{P}(\mathbf{w}_i) \propto \exp\left[-\frac{1}{2}\sum_{j=1}^C g(w_i^{[j]}) \right] = \prod_{j=1}^C \exp\left[-\frac{1}{2}g(w_i^{[j]}) \right] = \prod_{j=1}^C \mathcal{P}(w_i^{[j]})$, with $g(w_i^{[j]})$ being a given function of $w_i^{[j]}$. Generally, $\mathbf{w}$ in (9) is sparse, and therefore certain sparsity properties should be enforced on $\mathbf{w}$. To this effect, the function $g(\cdot)$ is usually chosen to be a concave, non-decreasing function of $|w_i^{[j]}|$ [6]. Examples of such functions $g(\cdot)$ include Generalised Gaussian priors and Student's *t* priors (see [6], [14] for details).

Computing the posterior mean $\mathbb{E}(\mathbf{w}|\mathbf{y})$ is typically intractable because the posterior $\mathcal{P}(\mathbf{w}|\mathbf{y})$ is highly coupled and non-Gaussian. To alleviate this problem, ideally one would like to approximate $\mathcal{P}(\mathbf{w}|\mathbf{y})$ as a Gaussian distribution for which efficient algorithms to compute the posterior exist [13]. For this, the introduction of lower bounding *super-Gaussian* priors $\mathcal{P}(w_i^{[j]})$, i.e., $\mathcal{P}(w_i^{[j]}) = \max_{\gamma_i > 0} \mathcal{N}(w_i^{[j]}|0, \gamma_i)\varphi(\gamma_i)$, can be used to obtain an analytical approximation of $\mathcal{P}(\mathbf{w}|\mathbf{y})$ [14].

Note that problem (9) has a block-wise structure, i.e. the solution $\mathbf{w}$ is expected to be block-wise sparse. Therefore, sparsity promoting priors should be specified for $\mathcal{P}(\mathbf{w}_i), \ \forall i$. To do this, for each block $\mathbf{w}_i$, we define a hyper-parameter γ_i such that

$$
\begin{aligned}
\mathcal{P}(\mathbf{w}_i) &= \max_{\gamma_i > 0} \mathcal{N}(\mathbf{w}_i|\mathbf{0}, \gamma_i \mathbf{I}_C)\varphi(\gamma_i) \\
&= \max_{\gamma_i > 0} \prod_{j=1}^C \mathcal{N}(w_i^{[j]}|0, \gamma_i)\varphi(\gamma_i),
\end{aligned} \tag{12}
$$

where $\varphi(\gamma_i)$ is a nonnegative function, which is treated as a hyperprior with γ_i being its associated hyperparameter. Throughout, we call $\varphi(\gamma_i)$ the "*potential function*". This Gaussian relaxation is possible if and only if $\log \mathcal{P}(\sqrt{w_i})$ is concave on $(0, \infty)$. Defining

$$
\begin{aligned}
\boldsymbol{\gamma}_i &= [\gamma_i, \ldots, \gamma_i] \in \mathbb{R}^C, \quad \boldsymbol{\Gamma}_i = \text{diag}\left[\boldsymbol{\gamma}_i \right], \\
\boldsymbol{\gamma} &= [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N] \in \mathbb{R}^{NC}, \quad \boldsymbol{\Gamma} = \text{diag}\left[\boldsymbol{\gamma} \right],
\end{aligned} \tag{13}
$$

we have

$$\mathcal{P}(\mathbf{w}) = \prod_{i=1}^{N} \mathcal{P}(\mathbf{w}_i) = \max_{\boldsymbol{\gamma}>0} \mathcal{N}(\mathbf{w}|\mathbf{0}, \boldsymbol{\Gamma})\varphi(\boldsymbol{\gamma}). \qquad (14)$$

B. Cost Function

Once we introduce the Gaussian likelihood in (11) and the variational prior in (14), we can get the following optimisation problem jointly on $\mathbf{w}$, $\boldsymbol{\gamma}$ and $\mathbf{S}$.

Proposition 1: The unknowns $\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}$ can be obtained by solving the following optimisation problem

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}) = \min_{\mathbf{w},\boldsymbol{\gamma},\mathbf{S}} \{-\log|\mathbf{S}| + \log|\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^{-1} + \mathbf{A}^\top\mathbf{S}\mathbf{A}|$$
$$+ (\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}\,(\mathbf{y} - \mathbf{A}\mathbf{w}) + \mathbf{w}^\top\boldsymbol{\Gamma}^{-1}\mathbf{w} + \sum_{j=1}^{N} p(\gamma_j)\},$$
$$(15)$$

where $\boldsymbol{\Gamma}$ is given in (13).

Proof: The derivation can be found in the SI S5. The proof mainly relies on using marginal likelihood maximisation. ∎

C. Algorithm

The cost function in (15) is convex in $\mathbf{w}$ and $\mathbf{S}$ but concave in $\boldsymbol{\Gamma}$. This non-convex optimisation problem can be formulated as a convex-concave procedure (CCCP). It can be shown that solving this CCCP is equivalent to solving a series of iterative convex optimisation programs, which converges to a stationary point [15]. Let

$$u(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}) \triangleq (\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}\,(\mathbf{y} - \mathbf{A}\mathbf{w}) + \mathbf{w}^\top\boldsymbol{\Gamma}^{-1}\mathbf{w} - \log\det\mathbf{S}$$

$$v(\boldsymbol{\gamma}, \mathbf{S}) \triangleq - \left[\log|\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^{-1} + \mathbf{A}^\top\mathbf{S}\mathbf{A}| + \sum_{j=1}^{N} p(\gamma_j) \right].$$

It is easy to check that $v(\boldsymbol{\gamma}, \mathbf{S})$ is a convex function with respect to $\boldsymbol{\gamma}$. Furthermore, $\log|\cdot|$ is concave in the space of positive semi-definite matrices. Since we adopt a super-Gaussian prior with potential function $\varphi(\gamma_j), \forall j$, as described in (12), a direct consequence is that $p(\gamma_j) = -\log\varphi(\gamma_j)$ is concave, and, therefore, $-p(\gamma_j)$ is convex [5].[2] Note that $u(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S})$ is jointly convex in $\mathbf{w}$, $\boldsymbol{\gamma}$ and $\mathbf{S}$, while $v(\boldsymbol{\gamma}, \mathbf{S})$ is jointly convex in $\boldsymbol{\gamma}$ and $\mathbf{S}$. As a consequence, the minimisation of the objective function can be formulated as a concave-convex procedure

$$\min_{\boldsymbol{\gamma}\succeq\mathbf{0},\mathbf{S}\succeq\mathbf{0},\mathbf{w}} u(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}) - v(\boldsymbol{\gamma}, \mathbf{S}). \qquad (16)$$

Since $v(\boldsymbol{\gamma}, \mathbf{S})$ is differentiable over $\boldsymbol{\gamma}$, the problem in (16) can be transformed into the following iterative convex optimisation problem

$$\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\mathrm{argmin}}\, u(\mathbf{w}, \boldsymbol{\gamma}^k, \mathbf{S}^k) \qquad (17)$$

$$\boldsymbol{\gamma}^{k+1} = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\mathrm{argmin}}\, u(\mathbf{w}^k, \boldsymbol{\gamma}, \mathbf{S}^k) - \nabla_{\boldsymbol{\gamma}} v(\boldsymbol{\gamma}^k, \mathbf{S}^k)^\top\boldsymbol{\gamma} \qquad (18)$$

$$\mathbf{S}^{k+1} = \underset{\mathbf{S}\succeq\mathbf{0}}{\mathrm{argmin}}\, u(\mathbf{w}^k, \boldsymbol{\gamma}^k, \mathbf{S}) - \nabla_{\mathbf{S}} v(\boldsymbol{\gamma}^k, \mathbf{S}^k)^\top\mathbf{S}. \qquad (19)$$

---

[2]In this paper, the prior is chosen as a Student's t prior thus $p(\gamma_j) = 1$.

Using basic principles in convex analysis, we then obtain the following analytic form for the negative gradient of $v(\boldsymbol{\gamma})$ at $\boldsymbol{\gamma}$ (using the chain rule):

$$\boldsymbol{\alpha}^k \triangleq -\nabla_{\boldsymbol{\gamma}} v(\boldsymbol{\gamma}, \mathbf{S}^k)^\top|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^k}$$
$$= \nabla_{\boldsymbol{\gamma}} \left[\log|\boldsymbol{\Gamma}^{-1} + \mathbf{A}^\top\mathbf{S}^k\mathbf{A}| + \log|\boldsymbol{\Gamma}| \right]$$
$$= \mathrm{diag}\{\left[-(\boldsymbol{\Gamma}^k)^{-1} + \mathbf{A}^\top\mathbf{S}^k\mathbf{A} \right]^{-1}\}\cdot\mathrm{diag}\{-(\boldsymbol{\Gamma}^k)^{-2}\}$$
$$+ \mathrm{diag}^{-1}\{\boldsymbol{\Gamma}^k\} \qquad (20)$$
$$= \underbrace{\left[\; \boldsymbol{\alpha}_{11}^k \;|\; \cdots \;|\; \boldsymbol{\alpha}_{1N}^k \;\right]}_{N \text{ Blocks}}$$
$$= [\; \underbrace{\alpha_{11}^k,\ldots,\alpha_{11}^k}_{C \text{ Elements}} \;|\; \cdots \;|\; \underbrace{\alpha_{1N}^k,\ldots,\alpha_{1N}^k}_{C \text{ Elements}} \;].$$

Therefore, the iterative procedures (17) and (18) for $\mathbf{w}^{k+1}$ and $\boldsymbol{\gamma}^{k+1}$, respectively, can be formulated as

$$\left[\mathbf{w}^{k+1}, \boldsymbol{\gamma}^{k+1}\right] = \underset{\boldsymbol{\gamma}\succeq\mathbf{0},\mathbf{w}}{\mathrm{argmin}} (\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}^k (\mathbf{y} - \mathbf{A}\mathbf{w})$$
$$+ \sum_{i=1}^{N} \left(\frac{\mathbf{w}_i^\top\mathbf{w}_i}{\gamma_i} + C\gamma_i\alpha_i^k \right). \qquad (21)$$

The optimal $\boldsymbol{\gamma}$ components are obtained as: $\gamma_i = \frac{\|\mathbf{w}_i\|_2}{\sqrt{C\alpha_i^k}}$. If $\boldsymbol{\gamma}$ is fixed, we have $\mathbf{w}^{k+1}$ by solving optimisation problem

$$\min_{\mathbf{w}} (\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}^k (\mathbf{y} - \mathbf{A}\mathbf{w}) + 2\sum_{i=1}^{N} \|\theta_i^k \cdot \mathbf{w}_i\|_2, \qquad (22)$$

where $\theta_i^k = C\alpha_i^k$. We can then inject this into the expression of γ_i, which yields

$$\gamma_i^{k+1} = \frac{\|\mathbf{w}_i^{k+1}\|_2}{\sqrt{C\alpha_i^k}}. \qquad (23)$$

After we get $\mathbf{w}^{k+1}$ and $\boldsymbol{\gamma}^{k+1}$, we can proceed with the optimisation iteration in (19):

$$\Lambda^k = -\nabla_{\mathbf{S}} v(\boldsymbol{\gamma}^k, \mathbf{S}^k)$$
$$= \nabla_{\mathbf{S}}\left(\log\det\left(\boldsymbol{\Gamma}^{-k} + \mathbf{A}^\top\mathbf{S}^k\mathbf{A}\right) \right) \qquad (24)$$
$$= \mathbf{A}(\boldsymbol{\Gamma}^{-k} + \mathbf{A}^\top\mathbf{S}^k\mathbf{A})^{-1}\mathbf{A}^\top.$$

Letting $\mathbf{Y}^{k+1} = (\mathbf{A}\mathbf{w}^{k+1} - \mathbf{y})\cdot(\mathbf{A}\mathbf{w}^{k+1} - \mathbf{y})^\top$, we can get an estimate of the inverse of covariance matrix $\mathbf{S}$ as:

$$\mathbf{S}^{k+1} = \underset{\mathbf{S}\succeq\mathbf{0}}{\mathrm{argmin}}\, \mathrm{Tr}\left(\mathbf{S}\mathbf{Y}^{k+1}\right) - \log\det\mathbf{S} + \mathrm{Tr}\left(\Lambda^k\mathbf{S}\right). \qquad (25)$$

Given $\boldsymbol{\gamma}^{k+1}$ in (23) and $\mathbf{S}^{k+1}$ in (25), we can then go back to (20) to update $\boldsymbol{\alpha}$ for the next iteration.

This above described iterative procedure for identification is summarised in Algorithm 1 below. Some further discussion can be found in SI Section S6.

D. ADMM Implementation

Essentially, Algorithm (1) consists of a reweighted Group Lasso algorithm (26) and a reweighted inverse covariance estimation algorithm (27). Algorithm (1) can be implemented using the Alternating Direction Method of Multipliers (ADMM) [19]. This ADMM parallelisation allows to

Algorithm 1 Nonlinear Identification Algorithm using Heterogeneous Datasets

---

1: Collect C heterogeneous groups of time series data from the system of interest (assuming the system can be described by (1));
2: Select the candidate basis functions that will be used to construct the dictionary matrix described in Section III;

3: Initialise $\theta_i^0 = 1$, $\forall i$, $\alpha_i^0 = \frac{\theta_i^0}{C}$, $\mathbf{S}^0 = \mathbf{I}$, $\Lambda^0 = \mathbf{I}$;
4: **for** $k = 0, \ldots, k_{\max}$ **do**
5:    $\mathbf{w}^{k+1}$ can be obtained by solving the following weighted minimisation problem over $\mathbf{w}$, subject to the convex constraints in Assumption 1

$$\min_{\mathbf{w}} \frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}^k (\mathbf{y} - \mathbf{A}\mathbf{w}) + \sum_{i=1}^{N} \|\theta_i^k \cdot \mathbf{w}_i\|_2;$$

$$(26)$$

6:    Update γ_i^{k+1} using (23);
7:    Let $\mathbf{Y}^{k+1} = (\mathbf{A}\mathbf{w}^{k+1} - \mathbf{y}) \cdot (\mathbf{A}\mathbf{w}^{k+1} - \mathbf{y})^\top$;
8:    $\mathbf{S}^{k+1}$ can be obtained by solving the following weighted minimisation problem over the inverse of the covariace matrix:

$$\min_{\mathbf{S} \succeq \mathbf{0}} \text{Tr}\left(\mathbf{Y}^{k+1} + \Lambda^k\right)\mathbf{S} - \log \det \mathbf{S};$$

$$(27)$$

9:    Update $\boldsymbol{\alpha}^{k+1}$ using (20);
10:    Update $\theta_i^{k+1} = C\alpha_i^{k+1}$;
11:    Update Λ^{k+1} using (24);
12:    **if** a stopping criterion is satisfied **then**
13:        Break;
14:    **end if**
15: **end for**

---

distribute the algorithmic complexity to different threads and to build a platform for scalable distributed optimisation. This is key to be able to deal with problems of large dimensions. More details can be found in SI Section S7.

E. Connection to SDP formulations and the sparse multiple kernel method

The iteration in (21) can be rewritten in the following compact form

$$\left[\mathbf{w}^{k+1}, \boldsymbol{\gamma}^{k+1}\right] = \underset{\boldsymbol{\gamma} \succeq \mathbf{0}, \mathbf{w}}{\arg\min} \, (\mathbf{y} - \mathbf{A}\mathbf{w})^\top \mathbf{S}^k (\mathbf{y} - \mathbf{A}\mathbf{w})$$
$$+ \mathbf{w}^\top \boldsymbol{\Gamma}^{-1} \mathbf{w} - \nabla_{\boldsymbol{\gamma}} v(\boldsymbol{\gamma}^k, \mathbf{S}^k)^\top \boldsymbol{\gamma}.$$
$$(28)$$

This is equivalent to the following SDP by using the standard procedure in [17].

$$\min_{\mathbf{z}, \mathbf{w}, \boldsymbol{\gamma}} \quad \mathbf{z} - \nabla_{\boldsymbol{\gamma}} v(\boldsymbol{\gamma}^k, \mathbf{S}^k)^\top \boldsymbol{\gamma}$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{z} & (\mathbf{y} - \mathbf{A}\mathbf{w})^\top & \mathbf{w}^\top \\ \mathbf{y} - \mathbf{A}\mathbf{w} & (\mathbf{S}^k)^{-1} & \mathbf{0} \\ \mathbf{w} & \mathbf{0} & \boldsymbol{\Gamma} \end{bmatrix} \succeq \mathbf{0}$$
$$\boldsymbol{\gamma} \succeq \mathbf{0}$$

The cost of solving this SDP is at least N^3 as well as M. Therefore, solving this SDP is too costly for all but problems with a small number of variables. This means that

the number of samples, the dimension of the system, etc., can not be too large simultaneously. In this SDP formulation, $\boldsymbol{\Gamma}$ is closely related to the sparse multiple kernel presented in [18]. Certain choice of kernels may introduce some good properties or help reduce algorithmic complexity. In our case, we choose $\boldsymbol{\Gamma}$ to have a diagonal or a DC kernel structure.

V. SIMULATIONS

In this section, we use numerical simulations to show the effectiveness of the proposed algorithm. To compare the identification accuracy of the various algorithms considered, we use the root of normalised mean square error (RNMSE) as a performance index, i.e. **RNMSE** $= \|\mathbf{w}_{\text{estimate}} - \mathbf{w}_{\text{true}}\|_2 / \|\mathbf{w}_{\text{true}}\|_2$. Several factors affect the RNMSE, e.g. number of experiments C, measurement noise intensity, dynamic noise intensity, length of single time series data M, number of candidate basis functions N. For brevity of exposition, we only show results pertaining to change of RNMSE over number of experiment C and length of single time series for one experiment, all in the noiseless case. More results related to other factors that may affect RNMSE will be shown in a future journal publication presenting these results in more details.

As an illustrative example, we consider a model of an eight species generalised repressilator [20], which is a system where each of the species represses another species in a ring topology. The corresponding dynamic equations are as follows:

$$\dot{x}_{1t} = \frac{p_{11}}{p_{12}^{p_{13}} + x_{8t}^{p_{13}}} + p_{14} - p_{15}x_{1t},$$
$$\dot{x}_{it} = \frac{p_{i1}}{p_{i2}^{p_{i3}} + x_{i-1,t}^{p_{i3}}} + p_{i4} - p_{i5}x_{it}, \; \forall i = 2, \ldots 8,$$
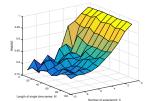$$(29)$$

where p_{ij}, $i = 1, \ldots, 8$, $j = 1, \ldots, 5$. We assume the mean value for these parameters across different species and experiments are $\bar{p}_{i1} = 40$, $\bar{p}_{i2} = 1$, $\bar{p}_{i3} = 3$, $\bar{p}_{i4} = 0.5$, $\bar{p}_{i5} = 1$, $\forall i$. We simulate the ODEs in (29) to generate the time series data. In each "experiment" or simulation of (29), the initial conditions are randomly drawn from a standard uniform distribution on the open interval $(0, 1)$. The parameters in each experiment vary no more than 20% of the mean values. In MATLAB, one can use $\bar{p}_{ij} * (0.8 + 0.4 * \text{rand}(1))$ to generate the corresponding parameters for each experiment.
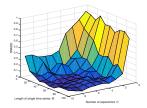
The numerical simulation procedure can be summarised as follows:

1) The deterministic system of ODEs (29) is solved numerically with an adaptive fourth-order Runge-Kutta method;
2) As explained in (6), Gaussian measurement noise with variance σ^2 is added to the corresponding time-series data obtained in the previous step[3];
3) The data is re-sampled with uniform intervals[4];
4) The local polynomial regression framework in [21] is applied to estimate the first derivative;

---

[3] In the example presented here, we consider the noiseless case corresponding to $\sigma = 0$.
[4] In this example, interval length is set to 1.

(a) Group Lasso (first iteration of Algorithm 1). The minimal RN-MSE is around 0.75

(b) Algorithm 1 with maximal iteration number $k_{\max} = 5$. The minimal RNMSE is almost 0.

Fig. 1. Algorithm comparison in terms of RNMSE averaged over 50 independent experiments. An enlarged version can be found in SI Section S8.

5) A dictionary matrix is constructed as explained in Section III;
6) Algorithm 1 is used to identify the model.

Following the procedure described in Section II, the candidate dictionary matrix $\mathbf{A}$ in step 5) above is constructed by selecting as candidate nonlinear basis functions typically used to represent terms appearing in ODE models of Gene Regulatory Networks. As a proof of concept, we only consider Hill functions as potential nonlinear candidate functions. The set of Hill functions with Hill coefficient h, both in activating and repressing form, for the ith state variables at time instant t_k are:

$$\text{hill}(x_{it}, K, h_{\text{num}}, h_{\text{den}}) \triangleq \frac{x_{it}^{h_{\text{num}}}}{K^{h_{\text{den}}} + x_{it}^{h_{\text{den}}}} \qquad (30)$$

where h_{num} and h_{den} represent the Hill coefficients. When $h_{\text{num}} = 0$, the Hill function has a repression form, whereas an activation form is obtained for $h_{\text{num}} = h_{\text{den}} \neq 0$.

In our identification experiment, we assume h_{num}, h_{den} and K to be known. We are interested in identifying the regulation type (linear or Hill type, repression or activation) and the corresponding parameters p_{i1}, the degradation rate constant p_{i4} and the basal expression rate $p_{i5}, \forall i$. Since there are 8 state variables, we can construct the dictionary matrix $\mathbf{A}$ with 8 (basis functions for linear terms) $+(2*8)$ (basis functions for Hill functions, both repression and activation form) $+1$ (constant unit vector) $= 25$ columns. The corresponding matrix $\mathbf{A}$ is given in Eq. (S8.1) in Supporting Information Section S8.

For a fixed number of experiments C and length of single time series M, we compute the RNMSE over 50 simulations by varying initial conditions and parameters p_{ij}. The RNMSE over C and M are shown in Fig. 2(a) and Fig. 2(b), using both group Lasso and Algorithm 1 with the maximal iteration number $k_{\max} = 5$ (see line 4 in Algorithm 1). Inspection of the results presented in Fig. 2(a) and Fig. 2(b) clearly show that Algorithm 1 outperforms significantly group Lasso in terms of RNMSE.

VI. DISCUSSIONS

There are several issues that we plan to further explore in the future. First, we are working on establishing the minimal sampling rate necessary to yield adequate numerical estimates of the first derivative matrix (see Eq. (2)). Second, further results not shown in this paper indicate that RNMSE is high when dynamic noise and measurement noise are high. We are currently working on finer characterisation of the "quality" of the identification in terms of the Signal-to-Noise ratio.

REFERENCES

[1] http://arxiv.org/abs/1509.05153
[2] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 1999.
[3] H.-M. Kaltenbach, S. Dimopoulos, and J. Stelling, "Systems analysis of cellular networks under uncertainty," *FEBS letters*, vol. 583, no. 24, pp. 3923–3930, 2009.
[4] J. Vanlier, C. Tiemann, P. Hilbers, and N. van Riel, "Parameter uncertainty in biochemical models described by ordinary differential equations," *Mathematical biosciences*, vol. 246, no. 2, pp. 305–314, 2013.
[5] M. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
[6] D. Wipf, B. Rao, and S. Nagarajan, "Latent variable bayesian models for promoting sparsity," *Information Theory, IEEE Transactions on*, vol. 57, no. 9, pp. 6236–6255, 2011.
[7] N. T. Ingolia and J. S. Weissman, "Systems biology: reverse engineering the cell," *Nature*, vol. 454, no. 7208, pp. 1059–1062, 2008.
[8] S. Boyd, L. El Ghaoul, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Society for Industrial Mathematics, 1987, vol. 15.
[9] R. Horn and C. Johnson, *Matrix analysis*. Cambridge university press, 1990.
[10] V. Cerone, D. Piga, and D. Regruto, "Enforcing stability constraints in set-membership identification of linear dynamic systems," *Automatica*, vol. 47, no. 11, pp. 2488–2494, 2011.
[11] M. Zavlanos, A. Julius, S. Boyd, and G. Pappas, "Inferring stable genetic networks from steady-state data," *Automatica*, vol. 47, no. 6, pp. 1113–1122, 2011.
[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial Mathematics, 1994.
[13] C. Bishop, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 4.
[14] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational EM algorithms for non-Gaussian latent variable models," *Advances in neural information processing systems*, vol. 18, p. 1059, 2006.
[15] B. K. Sriperumbudur and G. R. Lanckriet, "On the convergence of the concave-convex procedure." in *NIPS*, vol. 9, 2009, pp. 1759–1767.
[16] A. Aravkin, J. V. Burke, A. Chiuso, and G. Pillonetto, "Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ard and glasso," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 217–252, 2014.
[17] S. Boyd and L. Vandenberghe, *Convex optimisation*. Cambridge university press, 2004.
[18] T. Chen, M. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *Automatic Control, IEEE Transactions on*, vol. 59, no. 11, pp. 2933–2945, 2014.
[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
[20] N. Strelkowa and M. Barahona, "Switchable genetic oscillator operating in quasi-stable mode," *Journal of The Royal Society Interface*, p. rsif20090487, 2010.
[21] K. De Brabanter, J. De Brabanter, B. De Moor, and I. Gijbels, "Derivative estimation with local polynomial fitting," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 281–301, 2013.
[22] Y. Chang, R. Dobbe, P. Bhushan, J. W. Gray, C. J. Tomlin, "Retrieving common dynamics of gene regulatory networks under various perturbations," in *Proceeding of Conference on Decision and Control*, 2015.

Supporting Information

A. Notation

Notation: The notation in this paper is standard. Bold symbols are used to denote vectors and matrices. For a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{A}_{i,j} \in \mathbb{R}$ denotes the element in the ith row and j^{th} column, $\mathbf{A}_{i,:} \in \mathbb{R}^{1 \times N}$ denotes its ith row, $\mathbf{A}_{:,j} \in \mathbb{R}^{M \times 1}$ denotes its jth column. For a column vector $\boldsymbol{\alpha} \in \mathbb{R}^{N \times 1}$, α_i denotes its ith element. In particular, $\mathbf{I}_L$ denotes the identity matrix of size $L \times L$. We simply use $\mathbf{I}$ when the dimension is obvious from context. $\|\mathbf{w}\|_1$ and $\|\mathbf{w}\|_2$ denote the ℓ_1 and ℓ_2 norm of the vector $\mathbf{w}$, respectively. $\|\mathbf{w}\|_0$ denotes the ℓ_0 norm of the vector $\mathbf{w}$, which counts the number of nonzero elements in the vector $\mathbf{w}$. $\mathrm{diag}\,[\gamma_1, \ldots, \gamma_N]$ denotes a diagonal matrix with principal diagonal elements being $\gamma_1, \ldots, \gamma_N$. $\mathbb{E}(\boldsymbol{\alpha})$ stands for the expectation of the stochastic variable $\boldsymbol{\alpha}$. $\propto$ means "proportional to". $\mathrm{blkdiag}[\mathbf{A}^{[1]}, \ldots, \mathbf{A}^{[C]}]$ denotes a block diagonal matrix with principal diagonal blocks being $\mathbf{A}^{[1]}, \ldots, \mathbf{A}^{[C]}$ in turn. $\mathrm{Tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$. A matrix $\mathbf{A} \succeq \mathbf{0}$ means $\mathbf{A}$ is positive semidefinite. A vector $\boldsymbol{\gamma} \succeq \mathbf{0}$ means each element in $\boldsymbol{\gamma}$ is nonnegative.

B. Discussion on Selection of Basis Fucntions

Some *a priori* knowledge of the field for which the models are developed can be helpful. Indeed, depending on the field for which the dynamical model needs to be built, only a few typical nonlinearities specific to this field need to be considered. For example, the class of models that arise from biochemical reaction network typically involves nonlinearities that capture fundamental biochemical kinetic laws, e.g. first-order functions $f([S]) = [S]$, mass action functions $f([S_1], [S_2]) = [S_1] \cdot [S_2]$, Michaelis-Menten functions $f([S]) = V_{\max}[S]/(K + [S])$, or Hill functions $f([S]) = V_{\max}[S]^h/(K^h + [S]^h)$.

C. Estimation of the First Derivative

Estimating time derivatives from noisy data in continuous-time systems can either be achieved using a measurement equipment with a sufficiently high sampling rate, or by using state-of-the-art mathematical approaches [21]. Estimation of derivatives is key to the identification procedure [21]. As pointed out in [1], the identification problem is generally solved through discretisation of the proposed model. Assuming that samples are taken at sufficiently short time intervals, various discretisation methods can be applied. Typically, a forward Euler discretisation is used to approximate first derivatives, i.e. $\mathbf{y}_i$ can be defined as

$$\mathbf{y}_i \triangleq \left[\frac{x_{i2} - x_{i1}}{\Delta t}, \ldots, \frac{x_{i,M+1} - x_{iM}}{\Delta t}\right]^\top \in \mathbb{R}^{M \times 1}.$$

In this paper, the local polynomial regression framework in [21] is applied to estimate $\dot{\mathbf{x}}(t)$. Forward Euler discretisation and central difference discretisation are special cases of the local polynomial regression framework.

Proposition 2 (Proposition 1 in [21]): Consider the bivariate data $(t_1, Y_1), \ldots, (t_M, Y_M)$. Assume data are equispace-sampled and let $\sum_{j=1}^k w_j = 1$. For $k+1 \leq i \leq n-k$, the weights w_j are chosen as:

$$w_j = \frac{6j^2}{k(k+1)(k+2)}, \quad j = 1, \ldots, k. \tag{S3.1}$$

Based on these weights, the first derivative can be approximated as:

$$\dot{Y}_i = \sum_{j=1}^k w_j \cdot \left(\frac{Y_{i+j} - Y_{i-j}}{t_{i+j} - t_{i-j}}\right). \tag{S3.2}$$

D. Derivation of Correlated Noise

Without loss of generality and to ease notation, we consider the scalar case. In the scalar case, the system equation is written:

$$\dot{x}_t = g(x_t) + \eta_{x_t}, \tag{S4.1}$$

while the measurement equation is given as:

$$y_t = x_t + \epsilon_t, \tag{S4.2}$$

where ϵ_t is the measurement noise, which is assumed to be Gaussian i.i.d. We can simply use Taylor series expansion to expand $g(x_t)$ from y_t:

$$\begin{aligned}
g(x_t) &= g(y_t - \epsilon_t) \\
&= g(y_t) \underbrace{- g'(y)|_{y=y_t}\epsilon_t + \mathcal{O}(\epsilon^2(t))}_{\textbf{Correlated Gaussian noise}} \\
&= g(y_t) + \eta_{y_t}.
\end{aligned} \tag{S4.3}$$

Therefore, if we can estimate $\dot{x}$ from y properly, we can write the following

$$\dot{x}_{\textbf{estimate}}(t) = g(y(t)) + \underbrace{\eta_x(t) + \eta_y(t)}_{\textbf{new noise}}$$

$$= g(y(t)) + \eta(t).$$

(S4.4)

Clearly, $\eta(t)$ is not independent and identically distributed anymore.

E. Derivation of the Cost Function

To derive the cost function in Section IV-B, we first introduce the posterior mean and variance

$$\mathbf{m_w} = \mathbf{\Sigma_w} \mathbf{A}^\top \mathbf{S} \mathbf{y},$$

(S5.1)

$$\mathbf{\Sigma_w} = (\mathbf{A}^\top \mathbf{S} \mathbf{A} + \mathbf{\Gamma}^{-1})^{-1}.$$

(S5.2)

Since the data likelihood $\mathcal{P}(\mathbf{y}|\mathbf{w})$ is Gaussian,

$$\mathcal{N}(\mathbf{y}|\mathbf{Aw}, \mathbf{S}^{-1})$$
$$= \frac{1}{(2\pi)^{M/2} |\mathbf{S}|^{-1/2}} \exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{Aw})^\top \mathbf{S}(\mathbf{y} - \mathbf{Aw}) \right],$$

(S5.3)

we can write the marginal likelihood as

$$\int \mathcal{N}(\mathbf{y}|\mathbf{Aw}, \mathbf{\Pi}) \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Gamma}) \prod_{j=1}^{N} \varphi(\gamma_j) d\mathbf{w}$$
$$= \frac{1}{(2\pi)^{M/2} |\mathbf{S}|^{-1/2}} \frac{1}{(2\pi)^N} \int \exp\{-E(\mathbf{w})\} d\mathbf{w} \prod_{j=1}^{N} \varphi(\gamma_j),$$

(S5.4)

where

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{Aw})^\top \mathbf{S}(\mathbf{y} - \mathbf{Aw}) + \frac{1}{2}\mathbf{w}^\top \mathbf{\Gamma}^{-1} \mathbf{w}.$$

(S5.5)

Equivalently, we get

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{m_w})^\top \mathbf{\Sigma_w}^{-1}(\mathbf{w} - \mathbf{m_w}) + E(\mathbf{y}),$$

(S5.6)

where $\mathbf{m_w}$ and $\mathbf{\Sigma_w}$ are given by (S5.1) and (S5.2).

We first show the data-dependent term is convex in $\mathbf{w}$ and $\boldsymbol{\gamma}$. From (S5.1), (S5.2), the data-dependent term can be re-expressed as

$$\begin{aligned} E(\mathbf{y}) =& \frac{1}{2}\left(\mathbf{y}^\top \mathbf{S} \mathbf{y} - \mathbf{y}^\top \mathbf{S} \mathbf{A} \mathbf{\Sigma_w} \mathbf{A}^\top \mathbf{S} \mathbf{y}\right) \\ =& \frac{1}{2}\left(\mathbf{y}^\top \mathbf{S} \mathbf{y} - \mathbf{y}^\top \mathbf{S} \mathbf{A} \mathbf{\Sigma_w} \mathbf{\Sigma_w}^{-1} \mathbf{\Sigma_w} \mathbf{A}^\top \mathbf{S} \mathbf{y}\right) \\ =& \frac{1}{2}(\mathbf{y} - \mathbf{Am_w})^\top \mathbf{S}(\mathbf{y} - \mathbf{Am_w}) + \frac{1}{2}\mathbf{m_w}^\top \mathbf{\Gamma}^{-1} \mathbf{m_w} \\ =& \min_{\mathbf{w}} \left[\frac{1}{2}(\mathbf{y} - \mathbf{Aw})^\top \mathbf{S}(\mathbf{y} - \mathbf{Aw}) + \frac{1}{2}\mathbf{w}^\top \mathbf{\Gamma}^{-1} \mathbf{w} \right]. \end{aligned}$$

(S5.7)

Using (S5.6), we can evaluate the integral in (S5.4) to obtain

$$\int \exp\{-E(\mathbf{w})\} d\mathbf{w} = \exp\{-E(\mathbf{y})\}(2\pi)^N |\mathbf{\Sigma_w}|^{1/2}.$$

(S5.8)

Applying a $-2\log(\cdot)$ transformation to (S5.4), we have

$$-2\log\left[\frac{1}{(2\pi)^{M/2}|\mathbf{S}|^{-1/2}} \frac{1}{(2\pi)^N} \int \exp\{-E(\mathbf{w})\} d\mathbf{w} \prod_{j=1}^{N} \varphi(\gamma_j) \right]$$
$$=(M+N)\log 2\pi - \log|\mathbf{S}| + \log|\mathbf{\Gamma}| + \log|\mathbf{\Gamma}^{-1} + \mathbf{A}^\top \mathbf{S} \mathbf{A}|$$
$$+ \sum_{j=1}^{N} p(\gamma_j) + (\mathbf{y} - \mathbf{Aw})^\top \mathbf{S}(\mathbf{y} - \mathbf{Aw}) + \mathbf{w}^\top \mathbf{\Gamma}^{-1} \mathbf{w}.$$

(S5.9)

Therefore we get the following cost function to be minimised in (15) over $\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}$

$$
\begin{aligned}
\mathcal{L}(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{S}) \\
= -\log|\mathbf{S}| + \log|\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^{-1} + \mathbf{A}^{\top}\mathbf{S}\mathbf{A}| \\
+ (\mathbf{y} - \mathbf{A}\mathbf{w})^{\top}\mathbf{S}(\mathbf{y} - \mathbf{A}\mathbf{w}) + \mathbf{w}^{\top}\boldsymbol{\Gamma}^{-1}\mathbf{w} + \sum_{j=1}^{N} p(\gamma_j).
\end{aligned}
$$

F. Some Discussion on Algorithm

Remark 3: It should be noted that when noise is Gaussian i.i.d. with <u>known</u> variance, sparse Bayesian learning algorithms are provably better than classic Group Lasso algorithms in terms of mean square error [16].

Remark 4: The initialisation step is important (line 3 of in Algorithm 1). In special cases where the process noise in (1) is Gaussian i.i.d and there is no measurement noise, $\mathbf{S}$ can be fixed to $\lambda^{-1}\mathbf{I}$ for all k, where λ is a positive real number, i.e. no update through (27) is carried out. In such situations, λ can be treated as the equivalent of the regularisation/trade-off parameter in a Group Lasso algorithm (27). In such cases, cross validation can be implemented through variations of the initialisation values.

Remark 5: When the model obtained is used for prediction purposes, the inverse covariance estimation procedure in (27) can be used for quantification of the prediction uncertainty or risk.

G. ADMM Implementation

Essentially, Algorithm (1) consists of a reweighted Group Lasso algorithm (26) and a reweighted inverse covariance estimation algorithm (27). Algorithm (1) can be implemented using the Alternating Direction Method of Multipliers (ADMM) [19]. This ADMM parallelisation allows to distribute the algorithmic complexity to different threads and to build a platform for scalable distributed optimisation. This is key to be able to deal with problems of large dimensions. ADMM can be used to obtain solutions to problems of the following form:

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & f(\mathbf{w}) + g(\mathbf{z}), \\
\text{subject to} \quad & P\mathbf{w} + Q\mathbf{z} = \mathbf{c},
\end{aligned}
\tag{S7.1}
$$

where $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$, $P \in \mathbb{R}^{p \times n}$, $Q \in \mathbb{R}^{p \times m}$, and $\mathbf{c} \in \mathbb{R}^p$. The functions $f(\cdot)$ and $g(\cdot)$ are convex, but can be nonsmooth, e.g. weighted ℓ_1 norm. The first step of the method consists in forming the augmented Lagrangian

$$
\begin{aligned}
L_\rho = & f(\mathbf{w}) + g(\mathbf{z}) + \mathbf{u}^{\top}(P\mathbf{w} + Q\mathbf{z} - c) + \\
& \rho/2\|P\mathbf{w} + Q\mathbf{z} - c\|_2^2.
\end{aligned}
\tag{S7.2}
$$

After that optimisation programmes with respect to different variables can be solved separately as follows:

$$
\begin{aligned}
\mathbf{w}^{\tau+1} & := \operatorname*{argmin}_{\mathbf{w}} \left(f(\mathbf{w}) + \frac{\rho}{2}\|P\mathbf{w} + Q\mathbf{z}^\tau - \mathbf{c} + \mathbf{u}^\tau\|_2^2 \right) \\
\mathbf{z}^{\tau+1} & := \operatorname*{argmin}_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2}\|P\mathbf{w}^{\tau+1} + Q\mathbf{z} - \mathbf{c} + \mathbf{u}^\tau\|_2^2 \right) \\
\mathbf{u}^{\tau+1} & := \mathbf{u}^\tau + P\mathbf{w}^{\tau+1} + Q\mathbf{z}^{\tau+1} - \mathbf{c}.
\end{aligned}
$$

If $g(z)$ is equal to $\lambda\|z\|_1$, then the update on z is simply

$$
\mathbf{z}^{\tau+1} = S_{\lambda/\rho}(P\mathbf{w}^{\tau+1} + \mathbf{u}^\tau),
$$

where $S_{\lambda/\rho}$ is the soft thresholding operator defined as

$$
S_{\lambda/\rho}(x) = \begin{cases} x - \lambda/\rho & \text{if } x > \lambda/\rho \\ 0 & \text{if } |x| < \lambda/\rho \\ x + \lambda/\rho & \text{if } x < -\lambda/\rho \end{cases}
$$

or

$$
S_{\lambda/\rho}(x) = \max(0, x - \lambda/\rho) - \max(0, -x - \lambda/\rho).
$$

Based on the above, we can design a simple algorithm to solve a nonsmooth optimisation problem in a decentralised fashion. Moreover, this algorithm converges provided the following stopping criterion is satisfied:
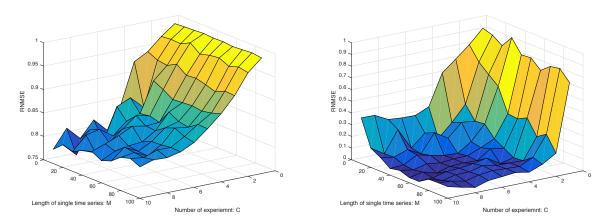
$$
\|\mathbf{w}^\tau - \mathbf{z}^\tau\|_2 \le \epsilon_{primal}, \quad \|\rho(\mathbf{z}^\tau - \mathbf{z}^{\tau-1})\|_2 \le \epsilon_{dual},
$$

where, the tolerances $\epsilon_{primal} > 0$ and $\epsilon_{dual} > 0$ can be set via an "absolute plus relative" criterion, e.g.

$$
\begin{aligned}
\epsilon_{primal} & = \sqrt{n}\epsilon_{abs} + \epsilon_{rel}\max(\|\mathbf{w}^\tau\|_2, \|\mathbf{z}^\tau\|_2), \\
\epsilon_{dual} & = \sqrt{n}\epsilon_{abs} + \epsilon_{rel}\rho\|\mathbf{u}^\tau\|,
\end{aligned}
$$

$$\mathbf{A} = \begin{bmatrix} x_{11} & \dots & x_{81} & \text{hill}(x_{11}, 1, 0, 3) & \dots & \text{hill}(x_{81}, 1, 3, 3) & 1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ x_{1M} & \dots & x_{8M} & \text{hill}(x_{1M}, 1, 0, 3) & \dots & \text{hill}(x_{8M}, 1, 3, 3) & 1 \end{bmatrix} \in \mathbb{R}^{M \times 25}. \tag{S8.1}$$

(a) Root of Normalised Mean Square Error averaged over 50 independent experiments using group Lasso algorithm (first iteration of Algorithm 1)

(b) Root of Normalised Mean Square Error averaged over 50 independent experiments using Algorithm 1 with maximal iteration number $k_{\max} = 5$.

Fig. S2. Algorithm comparison in terms of RNMSE.

where ϵ_{abs} and ϵ_{rel} are absolute and relative tolerances. More details can be found in [19].

More specifically, step (27) can be solved using ADMM instead:

$$\min_{\mathbf{w}} \quad (\mathbf{y} - \mathbf{A}\mathbf{w})^{\top} \mathbf{S}^k (\mathbf{y} - \mathbf{A}\mathbf{w}) + 2 \sum_{i=1}^{N} \|\mathbf{z}_i\|_2, \tag{S7.3}$$
$$\text{subject to} \quad \theta_i^k \mathbf{w}_i - \mathbf{z}_i = 0, \quad i = 1, \dots, N,$$

The optimisation programmes with respect to different variables can be solved separately as follows:

$$\mathbf{w}^{\tau+1} = \left(\mathbf{A}^{\top} \mathbf{S}^k \mathbf{A} + \rho \mathbf{I}\right)^{-1} \left(\mathbf{A}^{\top} \mathbf{S}^k \mathbf{y} + \rho \left(\mathbf{z}^k - \mathbf{u}^k\right)\right),$$
$$\mathbf{z}_i^{\tau+1} = \mathcal{S}_{\lambda/\rho} \left(\theta_i^k \mathbf{w}_i^{\tau+1} + \mathbf{u}^{\tau}\right), \quad i = 1, \dots, N,$$
$$\mathbf{u}^{\tau+1} = \mathbf{u}^{\tau} + \boldsymbol{\theta} \mathbf{w}^{\tau+1} - \mathbf{z}^{\tau+1}.$$

$\mathcal{S}$ is the vector soft thresholding operator $\mathcal{S}_{\kappa} : \mathbb{R}^C \to \mathbb{R}^C$

$$\mathcal{S}_{\kappa}(a) = (1 - \kappa/\|a\|)_{+} a, \tag{S7.4}$$

where $\mathcal{S}_{\kappa}(0) = 0$. This formula reduces to the scalar soft thresholding operator when a is a scalar. More details can be found in [19].

H. Simulation

In this section, we put the constructed dictionary matrix and two enlarged figures.

REFERENCES

[1] A. Papachristodoulou and B. Recht, "Determining interconnections in chemical reaction networks," in *American Control Conference, 2007. ACC'07.* IEEE, 2007, pp. 4872–4877.