A HYBRID PARAMETER ESTIMATION ALGORITHM FOR S-SYSTEM MODEL OF GENE REGULATORY NETWORKS

Jer-Nan Juang∗ and Wesson Wu†

The reconstruction of a gene regulatory network expressed in terms of a S-system model may be accomplished by a simple task of parameter estimation. Empirical data indicate that biological gene networks are sparsely connected and the average number of upstream-regulators per gene is less than two, implying that most of parameter variables in the S-system model are zero. It is thus desired to search for a parameter estimation algorithm that is capable of identifying the connectivity of the gene network and determining its reduced number of non-zero parameters. A hybrid algorithm is presented for identification and parameter estimation of gene network structure described by a S-system model. It combines an optimization process with a system identification method commonly used in the aerospace community. Constraint equations in a matrix form are formulated to deal with the steady state and the network connectivity conditions. The system parameter vector resides in the null space of the constraint matrix. The resulting network structure and system parameters are optimally tuned by minimizing the error of state time history. A numerical experiment is given to illustrate the hybrid parameter estimation algorithm.

INTRODUCTION

Advancement in DNA microarray technology allows scientists to simultaneously measure the expression levels for thousands of genes over time. The resulting time series data of gene expression contain valuable information on the structure and dynamics of the underlying gene regulatory networks. However, this information is entirely implicit and requires computational methods for extraction and interpretation. Reconstruction of gene regulatory networks from time series gene expression data has become one of the most challenging tasks in systems biology.

The reconstruction of a gene regulatory network starts with choosing a mathematical model to represent the gene regulatory network. Then the model’s parameters are estimated by fitting to the measured time series data of gene expression. These parameters are used to identify the structure and describe the dynamics of the gene regulatory network. In recent years, many mathematical models have been developed to reconstruct gene regulatory networks from time series data of gene expression.1 These models range from abstract binary Boolean network model2 to detailed continuous dynamic model using nonlinear ordinary differential equations.3 Each kind of model has its advantages and limitations.

∗Professor, Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan; Adjunct Professor, Texas A&M University, USA; Fellow AAS, AIAA, ASME
†Assistant Professor, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
Among all available models of gene regulatory networks, S-system is a very good candidate. S-system model is based on a set of nonlinear coupled differential equations in power-law formalism, which is capable of capturing virtually any complicated phenomenon of a gene regulatory network, including complex oscillations and even chaos. The regular structure of S-system model makes theoretical analyses of the steady state and sensitivity of a gene regulatory network possible and relatively simple. Most importantly, the parameters of the S-system explicitly represent the structural and regulatory features of a gene regulatory network. Therefore, the reconstruction of a gene regulatory network can be reduced to the simpler task of parameter estimation. However, the weakness of the S-system model is that a large number of parameters ($2N(N+1)$ where $N$ is the number of genes in the network) has to be estimated.

Many algorithms have been developed to estimate the parameters of the S-system either by estimating all the $2N(N+1)$ parameters together or decoupling the original parameter estimation problem into $N$ subproblems. However, most of them are computationally expensive heuristic search algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA), Evolutionary Computation (EC), Differential Evolution (DE), cooperative coevolutionary algorithm, evolutionary optimization, or multiobjective optimization. Typically, it takes several hours or days to estimate the parameters even when $N$ is only five. Since $N$ is usually very large for a gene regulatory network, it is needed to develop a computationally efficient method for estimating the parameters of the S-system.

Empirical data indicate that biological gene networks are sparsely connected and the average number of upstream-regulators per gene is less than two. A sparse, minimally connected, genetic architecture may be a fundamental design constraint shaping the evolution of gene network complexity. A novel two-step optimization process is introduced in this paper. The first step is to determine the gene network structure having the minimal connectivity. Linear equality equations are formulated to incorporate the linear steady-state and connectivity constraints into the optimization process. A null-space approach is introduced to reduce the number of parameter coefficients to be determined in the optimization process. Instead of using the state time history to search for an optimal set of S-system parameters, a sub-optimization process is proposed by minimizing the error of the estimated and simulated state-derivative time histories for any given set of S-system parameters. The estimated state-derivatives are calculated from a state-space model determined by a linear system identification algorithm that maps the input-output data. One objective of the sub-optimization process is to avoid numerical integrations and thus speed up the optimization process. Note that studied the robustness evaluation and modeling of biosystems by the observer/Kalman filter identification (OKID) algorithm. The second step is to use the identified network structure and its resulting sub-optimization parameters to finalize the optimal process by minimizing the error of the measured and simulated state histories. The proposed method is called a hybrid parameter estimation for gene regulatory networks described by a S-system model. The hybrid algorithm uses an optimization process in conjunction with a system identification method developed in the aerospace community. The hybrid estimation algorithm is illustrated by a numerical experiment.

**BASIC FORMULATION**

The general equations describing the temporal changes in a biochemical system can be formulated as a S-system model where the S refers to synergism and saturation of the investigated system. The S-system model is consistent with a number of specific features of bio-chemical systems as well as with observations that are common to other branches of biology. Non-linear differential equations
for a general S-system model are commonly expressed by

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{n_s} x_j^{g_{i,j}} - \beta_i \prod_{j=1}^{n_s} x_j^{h_{i,j}}; i = 1, 2, \cdots, n_s$$  \hspace{1cm} (1)$$

where $x$ is the state variable, and $n_s$ is the number of states. The quantities $\alpha_i$ and $\beta_i$ are multiplicative parameters (rate constants), and $g_{i,j}$ and $h_{i,j}$ are exponential parameters (kinetic orders). The parameters $\alpha_i$ and $g_{i,j}$ always refer to production or synthesis, whereas the parameters $\beta_i$ and $h_{i,j}$ always refer to degradation or loss. Taking log of the steady state equations yields

$$\log \alpha_i + \sum_{j=1}^{n_s} g_{i,j} \log x_j - \log \beta_i - \sum_{j=1}^{n_s} h_{i,j} \log x_j = 0$$  \hspace{1cm} (2)$$

for $j = 1, 2, \cdots, n_s$, or in matrix form The basic advantage of the S-system is that it has a linear matrix equation to describe its steady state behavior. It is thus wise to make good use of the steady state equation.

**Nonlinear Optimization With Equality Constraints**

In general, the state derivatives are unavailable. The parameters estimation requires equality and inequality constraints, i.e., some parameters must be positive. Some existing optimization software tools may be used for parameter estimation. Nevertheless, a good initial guess is desired for almost any optimization technique.

Let us define the parameter vector by

$$\theta_{n_p \times 1} = \begin{bmatrix} \theta_1^T & \theta_2^T & \cdots & \theta_{n_s}^T \end{bmatrix}^T; \quad n_p = 2n_s(n_s + 1)$$  \hspace{1cm} (3)$$

where

$$\theta_i = \begin{bmatrix} \log \alpha_i & g_{i,1} & \cdots & g_{i,n_s} & \log \beta_i & h_{i,1} & \cdots & h_{i,n_s} \end{bmatrix}^T$$  \hspace{1cm} (4)$$

for $i = 1, 2, \ldots, n_s$, and the constraint matrix by

$$\Gamma_{n_s \times n_p} = \begin{bmatrix} \gamma_{1 \times (2n_s+2)} & 0_{1 \times (2n_s+2)} & \cdots & 0_{1 \times (2n_s+2)} \\ 0_{1 \times (2n_s+2)} & \gamma_{1 \times (2n_s+2)} & \cdots & 0_{1 \times (2n_s+2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1 \times (2n_s+2)} & 0_{1 \times (2n_s+2)} & \cdots & \gamma_{1 \times (2n_s+2)} \end{bmatrix}$$  \hspace{1cm} (5)$$

where

$$\gamma_{1 \times (2n_s+2)} = \begin{bmatrix} 1 & \log x_1 \cdot \log x_{n_s} - 1 - \log x_1 \cdot \log x_{n_s} \end{bmatrix}$$  \hspace{1cm} (6)$$

Equality constraint equation becomes

$$\Gamma_{n_s \times n_p} \theta_{n_p \times 1} = 0_{n_s \times 1}$$  \hspace{1cm} (7)$$

The parameter vector $\theta_{n_p \times 1}$ may be determined by the optimization process that minimizes the error between the measured state time history $x$ and the optimized state time history $\hat{x}$, i.e.,

$$\theta = \min (\|x - \hat{x}\|/\|\hat{x}\|)$$  \hspace{1cm} (8)$$
where

\[
x = \begin{bmatrix}
x(t_1) & x(t_2) & \cdots & x(t_\ell)
\end{bmatrix}
\]

\[
\hat{x} = \begin{bmatrix}
\hat{x}(t_1) & \hat{x}(t_2) & \cdots & \hat{x}(t_\ell)
\end{bmatrix}
\]

subject to the equality constraint equation (7). For multiple experiments, both matrices \(x\) and \(\hat{x}\) are augmented in row wise to include all measured and optimized data.

Nonlinear Optimization Without Equality Constraints

The parameter vector is in the null space of the constraint matrix \(\Gamma_{n_s \times n_p}\). Taking singular value Decomposition (SVD) of the constraint matrix \(\Gamma_{n_s \times n_p}\) yields

\[
\Gamma_{n_s \times n_p} = U_{n_s \times n_s} \begin{bmatrix}
\sigma_{n_s \times n_s} & 0_{n_s \times (n_p - n_s)}
\end{bmatrix} \begin{bmatrix}
V_{n_p \times n_s} & V_0_{n_p \times (n_p - n_s)}
\end{bmatrix}^T
\Rightarrow \Gamma_{n_s \times n_p} V_0_{n_p \times (n_p - n_s)} = 0_{n_s \times (n_p - n_s)}
\]

(10)

The parameter vector \(\theta_{n_p \times 1}\) can be solved by

\[
\theta_{n_p \times 1} = V_0_{n_p \times (n_p - n_s)} c_{(n_p - n_s) \times 1}
\]

(11)

where the \(n_p \times (n_p - n_s)\) matrix \(V_0_{n_p \times (n_p - n_s)}\) contains the basis vectors of the null space and \(c_{(n_p - n_s) \times 1}\) is the parameter-coefficient vector to be determined by the optimization process that minimizes the error between the measured state time history \(x\) and the optimized state time history \(\hat{x}\)

\[
c_{(n_p - n_s) \times 1} = \min \left( \|x - \hat{x}\| / \|\hat{x}\| \right)
\]

(12)

Instead of using the parameter vector \(\theta_{n_p \times 1}\) in the optimization process, one may minimize the objective function with respect to the parameter-coefficient vector \(c_{(n_p - n_s) \times 1}\) which is smaller in size than the original parameter vector.

OPTIMIZATION PROCESS

A novel two-step optimization process is introduced in this paper. The first step is to use a sub-optimization process to determine the gene network structure with the minimal connectivity. The sub-optimization process avoids numerical integrations in the conventional/direct optimization process. It begins with using a linear system identification algorithm to determine a state-space model that maps the input-output data. The main objective of identifying the state-space model is to estimate the state derivatives of the S-system model. Instead of using the state time history to search for an optimal set of S-system parameters, the sub-optimization process is done by minimizing the error of the estimated and computed state-derivative time histories for any given set of S-system parameters. The second step is to use the identified gene structure and its corresponding sub-optimal parameters to finalize the optimization process that minimizes the error of the measured and computed state histories.

Identification of Gene Network Structure

This section is divided into three parts including estimation of state derivatives, sub-optimization process, and identification of network connectivity. The three parts must be done in order.
**Estimation of State Derivatives:** Parameter estimation for general s-system differential equations appears to be difficult because they have more parameters than equations that require enough data and/or multiple experiments. In addition, the S-system equations are stiff in the sense of numerical integration that make the optimization process impossible to reach its global minimum unless a good set of initial parameters are given.

Here we present a novel approach to avoid the numerical integration by first estimating the state derivatives. The Eigensystem Realization Algorithm (ERA)\textsuperscript{13–17} is used to identify a linear state space model by applying the ERA method with the measured state time history. The ERA model describes the map from input to output measurement. Let the output vector $y(k)$ be

$$y(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{ns}(k) \end{bmatrix} = Cz(k) + Du(k)$$

(13)

where $k = 0, 1, 2, \ldots, \ell - 1$ is the time index giving the state time history and $\ell$ is the length of the data. A discrete-time state-space model can be identified from the output time history $y(k)$ to yield

$$z(k + 1) = Az(k) + Bu(k)$$
$$y(k) = Cz(k) + Du(k)$$

(14)

The quantities $A$, $B$, $C$, and $D$ are system matrices. The input $u(k)$ is zero except the initial value $u(0) = 1$. The quantity $z(k)$ is the state vector that may not be identical in length to $y(k)$. We have assumed that a pulse input is given to the state-space model to generate the output vector $y(k)$ for $k = 0, 1, 2, \ldots, \ell$. The matrices $A$, $B$, $C$, and $D$ are identified by ERA with the assumption of zero initial condition $z(0) = 0$.

The discrete-time model can be converted to its corresponding continuous-time model,

$$\dot{z}(t) = A_c z(t) + B_c u(t)$$

(15)

where

$$A_c = \frac{1}{\Delta t} \ln(A)$$
$$B_c = [A - I]^{-1} A_c B$$

(16)

The subscript $c$ for state matrix $A_c$ and input matrix $B_c$ implies that they are in the continuous-time domain. State derivatives can then be determined by

$$\dot{y}(t) = \begin{bmatrix} \dot{x}_1(k) \\ \dot{x}_2(k) \\ \vdots \\ \dot{x}_{ns}(k) \end{bmatrix} = C\dot{z}(k) = CA_c z(k); \quad k > 0$$

(17)

Note that this conversion is not unique and sampling time interval $\Delta t$ is assumed a constant value. The initial time derivative is

$$\dot{y}(0) = \begin{bmatrix} \dot{x}_1(0) \\ \dot{x}_2(0) \\ \vdots \\ \dot{x}_{ns}(0) \end{bmatrix} = C\dot{z}(0) = CA_c z(0) + CB_c u(0) = CB_c$$

(18)
The initial state-derivative estimate is generally inaccurate and not recommended for use in the parameter estimation process.

**Sub-Optimalization Process:** Now, the identification process of gene network structure begins with using the steady-state constraint matrix shown in Eq. (5) and computing its null-space basis vectors. Since the key purpose of this step is to determine the connectivity of the gene network, the initial parameters for $g_{i,j}$ and $h_{i,j}$ are set to zero but not $\alpha_i$ and $\beta_i$. Apply an existing optimization software to determine the system parameters by minimizing the error between the estimated state derivatives and the optimized state derivatives. Search for a vector $c$ such that the resulting parameter vector $\theta$ minimizes the error of the state-derivative vector

$$c_{(n_p-n_s)\times1} = \min \left( \frac{\| \hat{x} - \hat{\dot{x}} \|}{\| \hat{x} \|} \right)$$

with

$$\dot{x} = \begin{bmatrix} \dot{x}(1) & \dot{x}(2) & \cdots & \dot{x}(\ell - 1) \end{bmatrix}$$
$$\hat{\dot{x}} = \begin{bmatrix} \hat{\dot{x}}(1) & \hat{\dot{x}}(2) & \cdots & \hat{\dot{x}}(\ell - 1) \end{bmatrix}$$

where $\dot{x}(k), k = 1, 2, \ldots, \ell - 1$ are the ERA-identified state derivatives. For multiple experiments, both matrices $\dot{x}$ and $\hat{\dot{x}}$ are augmented in row wise to include all measured and optimized data. The optimization process is sub-optimal because it minimizes the state-derivative error shown in Eq. (19) rather than the state error described in Eq. (12).

The advantage of this sub-optimization process is obvious because it does not involve any integration of the $s$-system differential equations. If the ERA-identified state-space model is accurate to fit the measured data, it is reasonable to assume that the sub-optimization process (Eq. (19)) should yield a good set of parameter estimates. Of course, it is anticipated that the process (Eq. (19)) is less sensitive to the initial guess than the one shown in Eq. (12).

**Identification of Network Connectivity:** Three tasks are proposed to determine the network connectivity. First, consider relatively small sub-optimal parameters as zero parameters. Second, augment the constraint matrix corresponding to the zero parameters and compute its null-space basis vectors. Third, repeat the sub-optimization process to determine the optimal system parameters.

Let the sub-optimal parameter vector $\theta^s_i$ be

$$\theta^s_i = [ \log \alpha^s_i \ g^s_{i1} \ \cdots \ g^s_{i_{ns}} \ \log \beta^s_i \ h^s_{i1} \ \cdots \ h^s_{i_{ns}} ]^T$$

for $i = 1, 2, \ldots, n_s$. The superscript $s$ signifies sub-optimal parameters. Some of the parameters $g_{ij}$ and $h_{ij}$ are relatively small that may be negligible. Setting those relatively small parameters (a total of $\mu$ parameters) to zero yields a constraint matrix augmented by adding rows such that

$$\Gamma_{(n_s+\mu)\times n_p}[\theta_{n_p\times1} = 0$$

where

$$\Gamma_{(n_s+\mu)\times n_p} = \begin{bmatrix} \Gamma_{n_s\times n_p} \\ e_1_{1\times n_p} \\ \vdots \\ e_{\mu1\times n_p} \end{bmatrix}$$
$$ek_{1\times n_p} = \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}$$
for $k = 1, 2, \ldots, \mu$. The unity 1 is located at the position in the parameter vector corresponding to the small $g_{ij}$ or $h_{ij}$ that is set to zero. The basis-vector matrix for the null space of the augmented matrix is $V_0 np \times (n_p - n_s - \mu)$ such that

$$\Gamma_{(n_s + \mu) \times np} V_0 np \times (n_p - n_s - \mu) = 0 \tag{24}$$

that, in turn, implies

$$\theta_{np \times 1} = V_0 np \times (n_p - n_s - \mu) c_{(n_p - n_s - \mu) \times 1} \tag{25}$$

After setting a total of $\mu$ parameters to zeros, repeat the sub-optimization process to compute a new set of sub-optimal parameters $\theta_{np \times 1}^s$. The initial coefficient $c_{(n_p - n_s - \mu) \times 1}^s$ for the repetitive step is calculated by

$$\theta_{np \times 1}^s = V_0 np \times (n_p - n_s - \mu) c_{(n_p - n_s) \times 1}^s \approx V_0 np \times (n_p - n_s - \mu) c_{(n_p - n_s - \mu) \times 1}^s$$

that yields

$$c_{(n_p - n_s - \mu) \times 1}^s = V_0^T np \times (n_p - n_s - \mu) \theta_{np \times 1}^s \approx V_0^T np \times (n_p - n_s - \mu) V_0 np \times (n_p - n_s) c_{(n_p - n_s) \times 1}^s$$

The initial parameter coefficient vector $c_{(n_p - n_s - \mu) \times 1}^s$ is then used in the sub-optimization process again to produce the sub-optimal parameter coefficient vector $c_{(n_p - n_s - \mu) \times 1}^{s\mu}$ and its corresponding parameter vector $\theta_{np \times 1}^{s\mu}$ where $\theta_{np \times 1}^{s\mu} = V_0 np \times (n_p - n_s) c_{(n_p - n_s) \times 1}^{s\mu}$. One question arises what is the best $\mu$ that gives the minimal sub-optimization error. To answer this question, an intuitive way is proposed as follows.

First, sort the resulting sub-optimal parameters $\theta_{i}^s$ shown in Eq. (21) and repeat the sub-optimization process with the smallest parameter set to zero. Then continue the sub-optimization process with the first few small parameters to zeros up to a reasonable limit. Finally, select the number $\mu$ of zero parameters that has the minimal optimization error among all repetitive sub-optimization runs. Note that any parameter $g_{i,j}$ (or $h_{i,j}$) is determined to be zero implying that the $i$th gene is not connected with the $j$th gene in synthesis (or degradation).

To this end, the connectivity of the S-system model describing the gene network structure has been determined by the optimal non-zero parameters. The non-zero parameters should give not only the network connectivity but also optimal values that approach to the actual parameters, if the estimated state-derivatives approach to their true values. One may stop the optimization process here without going further. If not, a fine-tuning optimization process is needed as shown in the following section.

**Fine-tuning Optimization Process**

The non-zero optimal parameters with minimal sub-optimization error are used to serve as the initial values for the fine-tuning optimal process as shown in Eq. (12). First, call the sub-optimal coefficient vector $c_{(n_p - n_s - \mu) \times 1}^{s\mu}$ and the null-space basis-vector matrix $V_0 np \times (n_p - n_s - \mu)$ that are obtained from the sub-optimization process with the minimal error between the identified and estimated state derivatives. Insert the sub-optimal coefficient vector $c_{(n_p - n_s - \mu) \times 1}^{s\mu}$ as the initial guess into the optimization process with the same upper- and lower-bounds as used in the sub-optimization process to
obtain the optimal coefficient vector \( \mathbf{c} \) that minimizes the error between the measured states and the estimated states integrated from the S-system equation. Use the optimal coefficient vector \( \mathbf{c} \) and the basis-vector matrix \( \mathbf{V}_{0}^{n_{p} \times (n_{p} - n_{s} - \mu)} \) for the null space of the augmented constraint matrix to compute the optimal system parameters, \( \mathbf{\theta}_{n_{p} \times 1} = \mathbf{V}_{0}^{n_{p} \times (n_{p} - n_{s} - \mu)} \mathbf{c} \).

NUMERICAL EXPERIMENT

A five-state s-system model is used to illustrate the optimization process developed in this paper. Table 1 shows all actual parameters for the five-state s-system model. Note that most of the parameters for \( g \) and \( h \) are zero, implying the network is sparsely connected. Table 2 shows ten sets of initial states for experimental simulations. The s-system differential equations are integrated using MATLAB function ode45 to generate a total of ten sets of data with five state time histories per set and the sampling interval at 0.003 unit.

<table>
<thead>
<tr>
<th>Actual Parameters</th>
<th>( g )</th>
<th>( h )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>( 5 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>( 10 )</td>
<td>( 2 )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>( 10 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>( 8 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>( 10 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
<th>9th</th>
<th>10th</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>0.12</td>
<td>0.7</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.7</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>0.14</td>
<td>0.7</td>
<td>0.14</td>
<td>0.14</td>
<td>0.7</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>4</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.7</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.7</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.7</td>
<td>0.18</td>
<td>0.18</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Identification of Gene Network Structure

The first 25 data points and the final steady-state data per set are used to go through the optimization process to estimate zero and non-zero system parameters. The first step is to identify a state-space model to give the input and output map of the state time history using the ERA method. Each data set (total 10 sets) is used to form an ERA-Hankel matrix of dimension \( 15 \times 23 \) that yields a state space model of order 10 described by the system matrices, \( A, B, C, \) and \( D \). The state-derivatives of the s-system can then be computed using Eq. (17) from the identified state-space model. For illustration, Fig. 1 shows the first 25 data points for the first set of time histories generated using the first set of initial states shown in Table 2, and its estimated state-derivative time
history. Note that the estimated state derivatives do not include the first data point because of its considerable sensitivity to the system uncertainty. Repeating the identification process for the other 9 sets of data yields the remaining sets of state-derivative time histories (not shown).

Most of optimization tools require to define a set of upper and lower bounds on the parameters. The upper bound $\theta^U_{60 \times 1}$ and lower bound $\theta^L_{60 \times 1}$ for the s-system parameters are $30 \geq \alpha_i, \beta_i \geq 1$ and $3 \geq g_{i,j}, h_{i,j} \geq -3$ for $i, j = 1, 2, \ldots, 5$. The upper and lower bounds must be in the null space of $V_{60 \times 55}$. Thus the upper bound for the coefficient vector $c^U_{55 \times 1}$ associated with the upper bound $\theta^U_{60 \times 1}$ must satisfy the following equality

$$\theta^U_{60 \times 1} = V_{60 \times 55} c^U_{55 \times 1} \Rightarrow c^U_{55 \times 1} = V_{60 \times 55}^T \theta^U_{60 \times 1} \quad (28)$$

Similarly, the lower bound for the coefficient vector $c^L_{55 \times 1}$ associated with the lower bound $\theta^L_{60 \times 1}$ must be related by

$$\theta^L_{60 \times 1} = V_{60 \times 55} c^L_{55 \times 1} \Rightarrow c^L_{55 \times 1} = V_{60 \times 55}^T \theta^L_{60 \times 1} \quad (29)$$

The corresponding upper and lower bounds for the coefficient vectors $c^U_{55 \times 1}$ and $c^L_{55 \times 1}$ are shown in Fig. 2.

With the state time histories and the estimated state-derivative time histories in hand, an optimization tool is used to compute the optimal vector $c_{55 \times 1}$ as defined in Eq. (19) that in turns yields the sub-optimal parameter vector $\theta_{60 \times 1}$ shown in Eq. (11). Matlab optimization function file "fmincon" with the sqp (Sequential Quadratic Programming) algorithm was used along with the initial
parameters shown in Table 3 to compute the sub-optimal parameters which are shown in Table 4. The initial parameters \( \alpha \) and \( \beta \) were randomly chosen using the Matlab function "randn" with "randn('seed',1)", i.e., \( \alpha = 10 \times \text{abs}(\text{randn}(5,1)) \) and \( \beta = 10 \times \text{abs}(\text{randn}(5,1)) \). The key objective of this step is to determine the zero parameters for \( g \) and \( h \) to establish the gene network structure. It is intuitive to set all initial parameters for \( g \) and \( h \) to zero. Table 4 clearly shows that many zero (or close to zero) parameters are identified and some small parameters are good candidates for further repetitive optimization processes.

Let the optimization process be repeated by setting some small parameters to zero and re-computing the optimal vector \( c_{(55-\mu) \times 1} \) as shown in Eq. (25) where \( \mu \) is the number of zero parameters. It is intuitive to start setting the smallest parameter to zero with \( \mu = 1 \), and repeat the process by setting the first two small parameters to zero with \( \mu = 2 \). The repetitive process continue until a specific number, say \( \mu = 37 \) is reached. At each repetition, a new constraint matrix \( \Gamma_{(5+\mu) \times 60} \)
shown in Eq. (23) with \( n_s = 5 \) and \( n_p = 60 \) is formed to compute its null-space basis-vector matrix \( V_{60 \times (55-\mu)} \). In addition, a new set upper and lower coefficient vectors must be re-computed by the following equations (similar to Eq. (27))

\[
\theta^U_{60 \times 1} \approx V_{60 \times (55-\mu)} c^U_{(55-\mu) \times 1}
\]

\[
\Rightarrow c^U_{(55-\mu) \times 1} \approx V^T_{60 \times (55-\mu)} V_{60 \times 55} c^U_{55 \times 1}
\]

and

\[
\theta^L_{60 \times 1} \approx V_{60 \times (55-\mu)} c^L_{(55-\mu) \times 1}
\]

\[
\Rightarrow c^L_{(55-\mu) \times 1} \approx V^T_{60 \times (55-\mu)} V_{60 \times 55} c^L_{55 \times 1}
\]

The approximation sign implies that the parameter vectors \( \theta^U_{60 \times 1} \) and \( \theta^L_{60 \times 1} \) may not be in the null space created by the columns of \( V_{60 \times (55-\mu)} \).

Performing the sub-optimization process for \( \mu = 1, 2, \ldots, 37 \) thus yields the optimization errors plotted in Fig. 3 which clearly indicates that the minimum takes place at \( \mu = 33 \). Figure 4 shows the sub-optimal coefficient vector \( c^S_{55 \times 1} \) (the top sub-figure), and its corresponding coefficient vector \( c^S_{22 \times 1} \) (the middle sub-figure) in the null space \( V_{60 \times 22} \) for \( \mu = 33 \), i.e., the first 33 small parameters were set to zero. Inserting \( c^S_{22 \times 1} \) as the initial coefficient vector for the identified network structure into the sub-optimization process yields the optimal coefficient vector \( c^{\mu}_{22 \times 1} \) (the bottom sub-figure). The coefficient upper and lower bounds for \( \mu = 33 \) are shown in Fig. 5. Although no significant discrepancy is observed between the coefficient vectors \( c^{\mu}_{22 \times 1} \) and \( c^S_{22 \times 1} \), the resulting non-zero sub-optimal parameters are noticeable as shown in Tables 4 and 5.

### Fine-tuning Optimization Process

If the sub-optimal results shown in Table 5 are not satisfied, the final step is to use the sub-optimal coefficient vector \( c^{\mu}_{22 \times 1} \) as the initial vector to fine-tune the S-system parameter estimation.

![Figure 3. Optimization error vs. number of zero parameters](image-url)
Figure 4. Parameter coefficient vector for identified network structure

Figure 5. Coefficient upper and lower bounds for identified network structure
Table 4. Sub-optimal parameters; Derivative-state error 0.64 %, State error 0.45%

<table>
<thead>
<tr>
<th>α</th>
<th>g</th>
<th>β</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>-0.04</td>
<td>0.102</td>
<td>0</td>
</tr>
<tr>
<td>8.81</td>
<td>2.16</td>
<td>-0.13</td>
<td>0</td>
</tr>
<tr>
<td>8.62</td>
<td>0</td>
<td>-1</td>
<td>-0.09</td>
</tr>
<tr>
<td>7.94</td>
<td>0</td>
<td>0</td>
<td>2.01</td>
</tr>
<tr>
<td>6.2</td>
<td>0.04</td>
<td>0</td>
<td>-0.21</td>
</tr>
</tbody>
</table>

Table 5. Sub-optimal parameters for identified network structure; Set the first 33 small parameters to aero; Derivative-state error 0.12 %, State error 0.05%

<table>
<thead>
<tr>
<th>α</th>
<th>g</th>
<th>β</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.02</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>7.96</td>
<td>0</td>
<td>0</td>
<td>2.01</td>
</tr>
<tr>
<td>9.76</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

by minimizing the error between the measured and computed state time histories. As used in the sub-optimal process, the same Matlab optimization function file "fmincon" with the sqp (Sequential Quadratic Programming) algorithm was used along with the null-space basis vector matrix \( V_{60 \times 22} \) and the upper and lower bounds shown in Fig. 5 to compute the final optimal parameters (see Table 6). Figure 6 shows sub-optimal coefficients and final/fine-tuning optimal coefficients with no significant discrepancy observed. However, checking Tables 5 and 6 reveals a significant improvement in state error (three-order smaller).

**CONCLUSION**

A hybrid parameter estimation method has been developed yielding a new optimization process for determination of gene network structures. A system identification method is used in conjunction with an off-the-shelf optimization technique to simultaneously determine the connectivity and pa-

Table 6. Optimal 60 Parameters; State error 0.000018%

<table>
<thead>
<tr>
<th>α</th>
<th>g</th>
<th>β</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
rameters of a gene network. The network parameter vector is found to reside in the null space of a constraint matrix containing the steady state equations and the network connectivity. In stead of directly identifying the network parameters, a smaller size of coefficient vector in length is calculated representing the parameter vector in the null space. The simulation example showed that the total computation time was 50 minutes on MacBook Air with a 1.86 GHz Intel core 2 Duo. The system identification part took less than 1 second, the sub-optimization process needed about 40 minutes, and the fine-tuning optimization process spent about 8 minutes.

**REFERENCES**


