2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Reverse-Engineering Biological Interaction Networks from Noisy Data using Regularized Least Squares and Instrumental Variables

Francesco Montefusco, Carlo Cosentino, Francesco Amato, Declan G. Bates

Abstract— **The problem of reverse engineering the topology of a biological network from noisy time–series measurements is one of the most important challenges in the field of Systems Biology. In this work, we develop a new inference approach which combines the Regularized Least Squares (RLS) technique with a technique to avoid the introduction of bias and non-consistency due to measurement noise in the estimation of the parameters in the standard Least Squares (LS) formulation, the Instrumental Variables (IV) method. We test our approach on a set of nonlinear *in silico* networks and show that the combined exploitation of RLS and IV methods improves the predictions with respect to other standard approaches.**

I. INTRODUCTION

A large number of conceptually different approaches have been proposed in the literature to reverse engineer biological systems at the molecular level using their measured responses to external perturbations (e.g. drugs, signalling molecules, pathogens) and changes in environmental conditions (e.g. change in the concentration of nutrients or in the temperature level). A significant difficulty for all of these approaches is the detrimental effect of measurement noise on the reliability of the inference results. Indeed, the performance of many current approaches has been shown to degrade significantly in the presence of even limited amounts of noise in the measurement data [1], [2].

While statistical methods are currently the most widely-used tools for network inference in biology, we are convinced that methods based on dynamical systems identification theory have great potential for application to network inference problems in Systems Biology. Approaches based on statistical models, such as Bayesian networks [3] and Mutual Information theory [4], usually require large data sets and/or assume that the samples are independent. In certain situations, however, only a small number of experimental data points may be available, and the assumption of independent samples is clearly not true when we consider the measurements of the expression of the same gene at two consecutive time–points. For such problems, the family of inference methods that use dynamical systems theory to identify linear models interpolating experimental data, [5], [6], [7], [1], [8], have been shown to be a useful alternative or complement to statistical approaches, especially when the size of the network to be reconstructed is moderate. In the recent literature, several authors have proposed techniques based on regression algorithms for the identification of linear models interpolating experimental data [9], [5], [6], [10], [7], [11].

A common limitation to the practical application of these methods, as noted above, is the detrimental effect of measurement noise. In a recent work, we addressed this issue by developing a novel approach, named PACTLS [12], which is devised to a) optimally deal with the presence of correlated noise in the measurements, by using the Constrained Total Least Squares (CTLS) algorithm (see [13]) and b) take into account qualitative prior knowledge about the network topology by representing this information as additional constraints for the reconstruction problem. While this approach appears to be highly promising, its computational complexity limits its application to relatively small-scale networks. In this work, we propose an alternative approach which allows us to explicitly take into account the effect of measurement noise within the inference process while minimising computational overheads so that much larger scale networks can be inferred. In order to avoid the introduction of bias and non-consistency due to measurement noise in the estimation of the parameters of a dynamical system by the Least Square (LS) method, we devised a new inference algorithm based on the use of Instrumental Variables (IV) [14], an extension of the standard LS. We investigated the possibility of using regularized techniques to deal with over-parameterized models and with the related problem of under-determination of the model structure. These techniques aim to reduce the model complexity by restricting the degrees of freedom in the model. As a result, only a small number of parameters (named effective parameters) are optimised, whereas the other spurious parameters are set to zero. In particular we combined the Regularized LS (RLS) technique, called ridge regressions in statistics, with the IV, to produce a novel approach named RLS-IV.

A statistical evaluation of the RLS-IV method has been performed by testing it over a set of *in silico* nonlinear networks, that have been generated by using GeneNetWeaver (GNW), an open-source tool for the automatic generation of *in silico* gene networks and reverse engineering benchmarks [15], [16], [17]. We compare the results with those obtained by the classical LS technique and by two extensions of the LS algorithm, an approach based on LS that only exploits the IV method, named LS-IV, and the RLS method or ridge regressions, in order to show that the combined exploitation of RLS and IV methods improves the predictions with

F. Montefusco and D.G. Bates are with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter EX4 4QF, UK {F.Montefusco, D.G.Bates}@exeter.ac.uk

C. Cosentino and F. Amato are with the School of Computer and Biomedical Engineering, Università degli Studi Magna Græcia di Catanzaro, Viale Europa, Campus di Germaneto, 88100 Catanzaro, Italy {carlo.cosentino, amato}@unicz.it

978-1-61284-801-3/11/$26.00 ©2011 IEEE

respect to other approaches based on LS.

The paper is structured as follows: Section II describes the network model and the generation of simulated data sets. Section III describes the novel inference technique proposed in this paper. The results obtained in the numerical tests are reported in Section IV. Finally, some conclusions and ideas for future extensions of our approach are given in Section V.

II. METHODS

A. Dynamical models for network inference

A standard approach to model the dynamics of biomolecular interaction networks is by means of a system of ordinary differential equations (ODEs) that describes the temporal evolution of the various compounds [18], [19]. Typically, the network is modeled as a system of rate equations in the form

$$\dot{x}_i(t) = f_i(x(t), p(t), u(t)), \tag{1}$$

for $i = 1, \ldots, n$ with $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, where the state variables x_i denote the quantities of the different compounds present in the system (e.g. mRNA, proteins, metabolites) at time t, f_i is the function that describes the rate of change of the state variable x_i and its dependence on the other state variables, p is the parameter set and u is the vector of external perturbation signals.

The level of detail and the complexity of these kinetic models can be adjusted, through the choice of the rate functions f_i, by using more or less detailed kinetics, i.e. specific forms of f_i (linear or specific types of nonlinear functions). Moreover, it is possible to adopt a more or less simplified set of entities and reactions, e.g. choosing whether to take into account mRNA and protein degradation, delays for transcription, translation or diffusion time [18].

When the order of the system increases, nonlinear ODE models quickly become intractable in terms of parametric analysis, numerical simulation and especially for identification purposes. Indeed, if the nonlinear functions f_i are allowed to take any form, determination of the network topology becomes impossible. Due to the above issues, although biomolecular networks are characterized by complex nonlinear dynamics, many network inference approaches are based on linear models or are limited to very specific types of nonlinear functions.

B. Generation of in silico nonlinear networks and data sets

The *in silico* networks used in this paper for benchmarking purposes have been generated by using GeneNetWeaver, an open-source tool for in silico benchmark generation and performance profiling of network inference methods [15], [16], [17]. The gene network is modelled by the following ODE system:

$$\begin{aligned} \frac{d[x_i]}{dt} &= m_i \cdot f_i(y) - \lambda_i^{RNA} \cdot x_i \\ \frac{d[y_i]}{dt} &= r_i \cdot x_i - \lambda_i^{Prot} \cdot y_i, \end{aligned}$$

where x_i and y_i are the mRNA and protein concentrations of every gene respectively, the m_i is the maximum transcription rate, r_i the translation rate, λ_i^{RNA} and λ_i^{Prot} are the mRNA and protein degradation rates, respectively. $f_i(\cdot)$ is the so-called input function of gene i, which determines the relative activation of the gene, modulated by the binding of transcription factors (TFs) to cis-regulatory sites, and is approximated using Hill-type terms. For our tests we used five networks of 10 nodes and used these networks to generate the datasets used in our statistical analysis of the performance of the different inference algorithms.

For each network we simulated time–courses showing how the network responds to a single perturbation consisting of the modification of the basal transcription rate of a single gene. For each experiment we generated a number of single perturbations, m, (corresponding to the number of time–series), equal to the number of nodes, n: the i-th perturbation consists of slightly increasing or decreasing the basal activation of the i-th node by a random amount. We simulated 50 experiments for each network and added noise to each simulation using the model of noise in microarrays [20], which is similar to a mix of normal and log-normal noise.

C. Linear model–based inference

Linear models are valid approaches for the network inference problem because, at least for small excursions of the relevant quantities from the equilibrium point, the dynamical evolution of almost all biological networks can be accurately described by means of linear systems, made up of ODEs in the continuous–time case, or difference equations in the discrete–time case (see [6], [10], [21], [8] and references therein).

Consider the continuous–time LTI model

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2}$$

where $x(t) = (x_1(t), \ldots, x_n(t))^T \in \mathbb{R}^n$, the state variables x_i, $i = 1, \ldots, n$, denote the quantities of the different compounds present in the system (e.g. mRNA concentrations for gene expression levels), $A \in \mathbb{R}^{n \times n}$ is the state transition matrix (the Jacobian of $f(x)$) and $B \in \mathbb{R}^{n \times m}$ is a matrix that determines the direct targets of external perturbations $u(t) = (u_1(t), \ldots, u_m(t))^T \in \mathbb{R}^m$ (e.g. drugs, overexpression or downregulation of specific genes), which are typically induced during *in vitro* experiments.

Note that the derivative (and therefore the evolution) of x_i at time t is directly influenced by the value $x_j(t)$ iff $A_{ij} \neq 0$. Moreover, the type (i.e. promoting or inhibiting) and extent of this influence can be associated with the sign and magnitude of the element A_{ij}, respectively. Thus, if we consider the state variables as quantities associated with the nodes of a network, the matrix A can be considered as a compact numerical representation of the network topology. Therefore, the topological reverse engineering problem can be recast as the problem of identifying the dynamical system (2). A possible criticism of this approach could be raised with respect to the use of a linear model, which is certainly inadequate to capture the complex nonlinear dynamics of certain molecular reactions. However, this criticism would

be reasonable only if the aim was to identify an accurate model of large changes in the states of a biological system over time, and this is not the case here. If the goal is simply to recover the qualitative functional relationships between the states of the system when the system is subjected to perturbations then a first–order linear approximation of the dynamics represents a valid choice of model.

III. RLS-IV INFERENCE ALGORITHM

A. Least Squares for dynamical systems

The basic step of the inference process consists of estimating, from time–course experimental data, the weighted connectivity matrix A and the exogenous perturbation matrix B of the *in silico* network model (2).

Assume that $h+1$ experimental observations, $x^{(j)}(k) \in \mathbb{R}^n$, $k = 0,\ldots,h$, are available, for each external perturbation $u^{(j)}(k)$, $j = 1,\ldots,m$. Then we can recast the problem in the discrete–time domain as

$$x^{(j)}(k+1) = Ax^{(j)}(k) + B_{\star,j}u^{(j)}(k), \tag{3}$$

where $B_{\star,j} \in \mathbb{R}^n$ is the j-th column of B. Let

$$Y^{(j)} = \begin{pmatrix} x_1^{(j)}(1) & \cdots & x_1^{(j)}(h) \\ \vdots & \ddots & \vdots \\ x_n^{(j)}(1) & \cdots & x_n^{(j)}(h) \end{pmatrix} \in \mathbb{R}^{n\times h},$$

$$X^{(j)} = \begin{pmatrix} x_1^{(j)}(0) & \cdots & x_1^{(j)}(h-1) \\ \vdots & \ddots & \vdots \\ x_n^{(j)}(0) & \cdots & x_n^{(j)}(h-1) \end{pmatrix} \in \mathbb{R}^{n\times h},$$

The identification model is then

$$\Xi := \begin{pmatrix} Y^{(1)} & \ldots & Y^{(m)} \end{pmatrix} = \Theta\Omega, \tag{4}$$

where

$$\Theta = \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}, \quad \Omega := \begin{pmatrix} X^{(1)} & \cdots & X^{(m)} \\ & I_m \otimes \mathbf{1}_{1\times h} & \end{pmatrix},$$

$I_m \in \mathbb{R}^{m\times m}$ is the identity matrix, $\mathbf{1} \in \mathbb{R}^{1\times h}$ is a vector of ones and $\otimes$ is the Kronecker product.

Each row, $\Theta_{i,\star}$, of the connectivity matrix Θ can be identified by using a multiple regression model

$$\Xi_i = Z \cdot \beta + \varepsilon, \tag{5}$$

where $\Xi_i = (Y_{i,\star}^{(1)},\ldots,Y_{i,\star}^{(m)})^T \in \mathbb{R}^{hm}$, $Y_{i,\star}^{(j)}, j = 1,\ldots,m$ is the i-th row of the $Y^{(j)}$ matrix, $Z = \Omega^T \in \mathbb{R}^{mh\times(n+m)}$, $\beta = \Theta_{i,\star}^T \in \mathbb{R}^{n+m}$ and $\varepsilon = (\varepsilon^{(1)}(1),\ldots,\varepsilon^{(1)}(h),\ldots,\varepsilon^{(m)}(1),\ldots,\varepsilon^{(m)}(h))^T \in \mathbb{R}^{mh}$ be the measurement noise. The (5) by the standard least squares (LS) method admits the following solution

$$\hat{\beta}_{LS} = \left(Z^T Z\right)^{-1} Z^T \Xi_i. \tag{6}$$

The matrix $\left(Z^T Z\right)^{-1} Z^T$ is called the (Moore-Penrose) pseudo-inverse of Z and is often denoted by $Z\dagger$. Note that, to compute $Z\dagger$, it is necessary that $Z^T Z$ is invertible; this is possible if the $n+m$ columns of Z (the regression vectors) are linearly independent, which requires $(m\times h) \geq n+m$, i.e., one should have at least as many measurements as regression coefficients. Note that the regressor matrix is not made up of independent variables: the columns of Z (the rows of Ω) are the state vectors at the steps $0,1,\ldots,h-1$, while the rows of Ξ are the same state vectors but shifted one step ahead. Thus, Ω and Ξ have $m\times(h-1)$ identical rows and differ only for m rows. A second point, which stems from the first, is that, in the LS formulation for dynamical system identification, the regressor variables are affected by noise: Z (or Ω^T) contains measured process outputs $x^{(j)}(k)$, that are non-deterministic owing to the noise, so the parameters are estimated *biased* and *non-consistent*. A bias means that the parameters systematically deviate from their optimal values; non-consistency means that the bias does not even approach zero as the number of data samples h goes to ∞. A final consideration concerns the correlation between the regressor columns of Z: examining (3) and considering a typical step response of a dynamical system, we can clearly see that the value of the state vector at the k-th step is dependent on the value at the previous step. If the dynamics of the system are smooth and slow, then $x^{(j)}(k)$ can be approximated by a linear combination of its values at the previous step, $x^{(j)}(k-1),\ldots,x^{(j)}(0)$. This fact is quite unfortunate, because it means the columns of $Z^T Z$ are almost linearly dependent, which produce a high sensitivity in the LS solution to noise and round-off errors.

We note that two standard strategies for improving the identification results when using time–series measurements, i.e. increasing the number of measurements by either reducing the sample time or by considering a longer time interval, are basically not useful in this context: indeed, having $x^{(j)}(k)$ too close in time to $x^{(j)}(k-1)$ just increases the approximate linear dependence between the regression vectors. On the other hand, if we carry on taking measurements after the signals have reached their steady–state values, this will again introduce new regression vectors that are linearly dependent on the others (the value of $x^{(j)}(k)$ is almost equal to $x^{(j)}(k-1)$). Hence, the only chance to improve the inference performance is by making many different experiments, possibly using different perturbation inputs which affect different nodes of the network.
Finally, note that, since the system evolution is sampled, $\hat{A}$ and $\hat{B}$ are not actually the estimates of A and B in (2), but rather of the corresponding matrices of the discrete–time system. We showed (see Appendix in [8]) that, if the sampling time is suitably small, in order to reconstruct the original sparsity pattern of the continuous–time system's matrices, one can set to zero the elements of the estimated matrices whose values are below a certain threshold.

B. Instrumental Variables Method

A promising approach to avoid the non-consistency of the parameter estimates, caused by the measurement noise, is the Instrumental Variables (IV) method (see [14], p. 486). We construct a matrix V that has the same dimensions of Z ($V \in \mathbb{R}^{mh\times(n+m)}$), whose columns are called *instrumental*

variables and are chosen to be uncorrelated with the noise, $V^T\varepsilon = 0$. Multiplying the (5) with V^T we have

$$V^T\Xi_i - V^TZ\beta = V^T\varepsilon = 0,$$

and consequently

$$V^T\Xi_i = V^TZ\beta\,. \tag{7}$$

The (7) admits the following solution:

$$\hat{\beta}_{LS-IV} = \left(V^TZ\right)^{-1}V^T\Xi_i\,. \tag{8}$$

The IV estimate (8) is equivalent to the estimate of the least square regression (6), if $V^T = Z^T$. Note that the columns of Z cannot be used as instrumental variables since Z is correlated with the noise ($Z^T\varepsilon \neq 0$).

In the following, a criterion which can be used in order to choose V is illustrated. The instrumental variables (IV) should be highly correlated with the regressors in order to make the variance error small. Good *IV* should be the measurements $x^{(j)}(k)$ without the noise. Then they can be approximated by filtering $x^{(j)}(k)$ through the process model and the following algorithm can be used:

P1) Estimate each row, $\Theta_{i,\star}$, of the connectivity matrix Θ by solving (6)

P2) Compute the evolution $\hat{x}^{(j)}(k)$ of the identified model Θ_{LS} in step $P1$).

P3) Construct V from the simulated data $\hat{x}^{(j)}(k)$:

$$V := \begin{pmatrix} \hat{X}^{(1)} & \cdots & \hat{X}^{(m)} \\ & I_m \otimes \mathbf{1}_{1\times h} & \end{pmatrix}^T,$$

where

$$\hat{X}^{(j)} = \begin{pmatrix} \hat{x}_1^{(j)}(0) & \cdots & \hat{x}_1^{(j)}(h-1) \\ \vdots & \ddots & \vdots \\ \hat{x}_n^{(j)}(0) & \cdots & \hat{x}_n^{(j)}(h-1) \end{pmatrix} \in \mathbb{R}^{n\times h},$$

for $j = 1,\ldots,m$.

P4) Solve (8) with the IV in step $P3$). Compute the data $\hat{x}_{IV}^{(j)}(k)$ obtained by the evolution of the identified model Θ_{LS-IV}. Update $\hat{x}^{(j)}(k)$ with $\hat{x}_{IV}^{(j)}(k)$ in $P3$).

P5) Compute the error matrix, defined as

$$E = \Xi - \Theta_{LS-IV}\Omega \in \mathbb{R}^{n\times mh}.$$

Define the residuals vector as

$$\begin{aligned} e_{IV} =& (e^{(1)}(1)^T,\ldots,e^{(1)}(h)^T \ldots \\ & e^{(m)}(1)^T,\ldots,e^{(m)}(h)^T)^T \in \mathbb{R}^{nmh}, \end{aligned}$$

where $e^{(j)}(k) = E_{\star,i} \in \mathbb{R}^n$ is the i-th column of E, where $i = j\cdot k$ for $k = 1,\ldots,h$ and for $j = 1,\ldots,m$.

P6) Construct an autoregressive (AR) model for the residuals to extract the remaining information from e_{IV}. For each $j = 1,\ldots,m$ we have the following model:

$$e^{(j)}(k) + a_1e^{(j)}(k-1) + \cdots + a_le^{(j)}(k-l) = v^{(j)}(k), \tag{9}$$

where $v^{(j)}(k) \in \mathbb{R}^n$ is the white noise and l is the dynamic order of the AR model. If we introduce the *backward shift operator* q^{-1} by

$$q^{-1}e^{(j)}(k) = e^{(j)}(k-1)$$

and

$$L(q) = 1 + a_1q^{-1} + \cdots + a_lq^{-l},$$

the (9) can be rewritten as

$$e^{(j)}(k) = \frac{1}{L(q)}v^{(j)}(k)\,.$$

By using the regression model in the form

$$X_e = \Lambda\cdot a + v,$$

where

$$\begin{aligned} X_e =& (e^{(1)}(h),\ldots,e^{(1)}(2),\ldots, \\ & e^{(m)}(h),\ldots,e^{(m)}(2))^T \in \mathbb{R}^{n(h-1)m} \\ v =& (v^{(1)}(h),\ldots,v^{(1)}(2),\ldots, \\ & v^{(m)}(h),\ldots,v^{(m)}(2))^T \in \mathbb{R}^{n(h-1)m} \end{aligned}$$

$a = (-a_1,\ldots,-a_l)^T \in \mathbb{R}^l$, and

$$\Lambda = \begin{pmatrix} Z_e^{(1)} \\ \vdots \\ Z_e^{(m)} \end{pmatrix} Z_e^{(j)} = \begin{pmatrix} e^{(j)}(h-1) & \cdots & e^{(j)}(h-l) \\ \vdots & \ddots & \vdots \\ e^{(j)}(h-l-1) & \cdots & e^{(j)}(1) \\ \vdots & \ddots & 0 \\ e^{(j)}(1) & 0 & 0 \end{pmatrix},$$

the optimal estimator for the vector a through the standard LS is given by

$$\hat{a}_{LS} = \left(\Lambda^T\Lambda\right)^{-1}\Lambda^TX_e\,.$$

P7) Filter the $\hat{x}_{IV}^{(j)}(k)$ computed in step $P4$) with the filter $\hat{L}(q)$ estimated in step $P6$):

$$\hat{x}_{i\,IV-L}^{(j)}(k) = \hat{L}(q)\hat{x}_{i\,IV}^{(j)}(k)\,, \quad i = 1,\ldots,n\,. \tag{10}$$

P8) Construct the matrix V^L from $\hat{x}_{IV-L}^{(j)}(k)$ obtained by (10):

$$V^L := \begin{pmatrix} \hat{X}_L^{(1)} & \cdots & \hat{X}_L^{(m)} \\ & I_m \otimes \mathbf{1}_{1\times h} & \end{pmatrix}^T,$$

where

$$\hat{X}_L^{(j)} = \begin{pmatrix} \hat{x}_{1\,IV-L}^{(j)}(0) & \cdots & \hat{x}_{1\,IV-L}^{(j)}(h-1) \\ \vdots & \ddots & \vdots \\ \hat{x}_{n\,IV-L}^{(j)}(0) & \cdots & \hat{x}_{n\,IV-L}^{(j)}(h-1) \end{pmatrix} \in \mathbb{R}^{n\times h},$$

for $j = 1,\ldots,m$.

P9) Solve (8) by using as *IV* the matrix V^L. Update $\hat{x}^{(j)}(k)$ in $P3$) with the new data obtained by the evolution of the identified model $\Theta_{LS-IV-L}$. Repeat steps $P3)-P9$) for three iterations (the procedure converges very fast).

C. Regularized Least Squares

The matrix $H = Z^T Z$ in (6) is identical to the Hessian of the loss function of the least squares problem (see [14], pp. 40–43) and it has to be well conditioned in order to obtain accurate parameter estimates. It is well known the probability of poor conditioning increases with the matrix dimension and increases the variance of the worst estimated parameters. The condition of the Hessian matrix, χ, can be defined by the eigenvalue spread of the matrix by the formula $\chi = \frac{\lambda_{max}}{\lambda_{min}}$, where λ_{max} and λ_{min} are the largest and smallest eigenvalue of H, respectively. A method for controlling the condition of the Hessian is called *ridge regression*. It consist of adding a factor α to all diagonal entries of the Hessian, $(Z^T Z + \alpha I)$. Then, the eigenvalues of H are changed, in particular the significant eigenvalues $(\lambda_i \gg \alpha)$ are not influenced by the addition of α, whereas the small eigenvalues $(\lambda_i \ll \alpha)$ are set to α. Therefore the condition of the Hessian can be directly controlled via $\chi_{reg} = \frac{\lambda_{max}}{\alpha}$, where χ_{reg} is the regularized eigenvalue spread. Then, to solve (5), the Regularized Least Squares (RLS) problem leads to the following parameter estimate

$$\hat{\beta}_{RLS} = \left(Z^T Z + \alpha I\right)^{-1} Z^T \Xi_i \,, \tag{11}$$

where $I \in \mathbb{R}^{mh \times mh}$ is the identity matrix. The method of generalized cross-validation (GCV) is used for choosing a good estimate of α from the data (see [22]).

Then the RLS is combined with IV. The estimation of each row of Θ is computed by solving the following formula:

$$\hat{\beta}_{RLS-IV} = \left(V^T Z + \alpha I\right)^{-1} V^T \Xi_i \,. \tag{12}$$

For choosing V, the same procedure illustrated above is used. For the estimation of the vector a of the AR filter defined by (9) the RLS approach is also used.

D. Edges selection

Independently from the chosen method used for the estimation of the connectivity matrix, all the elements of $\hat{A}$ are usually nonzero, whereas biological networks exhibit loose connectivity, that is, the number of connections per node is much lower than the total number of nodes. To evaluate the estimation of $\hat{A}$ in terms of network inference we have to normalise each element, by dividing it by the geometric mean of the norms of the row and column containing that element. Thus, we compute the normalised estimated adjacency matrix $\tilde{A}$, where

$$\tilde{A}_{ij} = \frac{\hat{A}_{ij}}{\left(\|\hat{A}_{\star,j}\| \cdot \|\hat{A}_{i,\star}\|\right)^{1/2}} \,. \tag{13}$$

To translate this estimated matrix into an inferred network, we sort the list of edges in descending order according to the absolute value of their corresponding estimated parameters. Then, the elements at the top of the list will correspond to high-confidence predictions, i.e., edges with high probability of actually existing in the original network. We also tested the cases where each element of $\hat{A}$ is normalized by dividing it only by the geometric mean of the row containing that element, or where the list of edges is generated by simply considering the absolute value of the $\hat{A}$. The best solution is obtained by using (13).

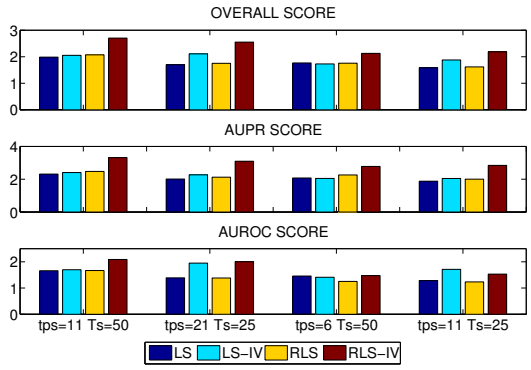

Fig. 1. Results for the LS approaches with four tests using different number of time points (tps) and of the sample time (Ts).

IV. RESULTS

The performance of the proposed algorithm has been evaluated by computing the AUPR index, which represents the area under the precision (or Positive Predictive Value, PPV) and recall (or sensitivity) curve (see [23], p.138, for computing the PPV and sensitivity indexes) and the AUROC index, that represents the area under the receiver operating characteristic (ROC) curve and summarizes the tradeoff between the true positive rate and the false positive rate. To compute these performance indexes, we do not consider the weight (sign) of an edge, but only its existence and direction.

The performance of the various methods based on LS is assessed by applying them to a set of 5 *in silico* nonlinear networks, each with ten nodes, generated by using GNW (see Section II-B). We assume that the perturbation targets and the qualitative effects of the perturbation are known, thus the pattern (but not the values of the non-zero elements) of $\hat{B}$ is preassigned. Each experiment, as explained above, consists of ten time–series (each one corresponding to a perturbation of a single node, $n = m$) and for each test we generated 50 experiments. Different tests have been conducted using different number of time points (tps) and of the sample time (Ts). To obtain a single performance measure for each test, we adopt the procedure used in [16] and compute the p-values for the AUPR and AUROC median values for each network. P-values for these assessments were obtained from the empirical distributions, estimated from 100,000 instances of random network link permutations. The overall AUPR p-value is computed by the geometric mean of the five AUPR p-values (denoted by $P1$) and the same procedure is used for computing the overall AUROC p-value (denoted by $P2$). Then a log-transformed average of the two overall AUROC and AUPR p-values, computed as $-0.5\log_{10}(P1 \cdot P2)$, gives us the OVERALL SCORE. Larger scores indicate greater statistical significance of the prediction and thus higher inference performance.

Fig. 1 shows the results of the different LS methods. We denote with 1) LS the approach based on standard

Least Squares, with 2) LS-IV the approach that exploits the iterative IV method, with 3) RLS the approach that uses the regularized technique, and with 4) RLS-IV the approach that combines the RLS with IV. The first column of Fig. 1 shows the OVERALL, AUPR and AUROC SCORES for the first test, using time–series data obtained by evolving the networks until most nodes reach a new steady–state ($tps = 11, Ts = 50$), and we can see that the best SCORE (OVERALL, AUPR and AUROC) is obtained by the RLS-IV approach. The other columns report the results obtained by the other tests. In particular, for the second column, we increase the tps by reducing the Ts, but the performance does not improve, in fact, as noted in Section. III-A, increasing the number of points by reducing the sample time is often not useful, because having the data points too close in time just increases the approximate linear dependence between the regression vectors, mainly in the last part of time–series. The last two columns of Fig. 1 show the results obtained by using the first part of the time–series, i.e. that containing the transient dynamics, and sampling with two different Ts. Also in this case, in terms of AUPR, the RLS-IV approach performs better than the others, whereas the AUROC SCORE is similar for all approaches. The performance, moreover, does not significantly degrade by decreasing the tps.

From the presented results, it is clear that the combined exploitation of the IV and RLS technique significantly improves the inference power of the standard LS algorithm. Note finally that in this paper, we have focussed our attention on network inference using only time–series data. If additional sources of data, such as steady–state data, are available, then the methods discussed in [16], [24] can also be used to further improve the inference capability.

V. CONCLUSIONS AND FUTURE WORKS

The results from numerical tests show that the RLS-IV approach, obtained by combing the Regularized LS technique and IV method, achieves a significantly improved inference capability. Compared to other approaches in the literature, the proposed approach efficiently minimizes computational overheads, and so in future work we will investigate the possibility of inferring large scale networks using this algorithm. This approach will then be combined with edge selection heuristics to exploit prior knowledge about some aspects of the network, in order to reconstruct large-scale networks with desired topologies, as illustrated for smaller-scale networks in our recent papers [8], [12].

VI. ACKNOWLEDGMENTS

This work was carried out under EPSRC Research Grant: EP/F057016/2.

REFERENCES

[1] C. Cosentino, W. Curatola, F. Montefusco, M. Bansal, D. di Bernardo, and F. Amato, "Linear matrix inequalities approach to reconstruction of biological networks," *IET Syst. Biol.*, vol. 1, no. 3, pp. 164–173, May 2007.

[2] M. Bansal and D. di Bernardo, "Inference of gene networks from temporal gene expression profiles," *IET Syst. Biol.*, vol. 1, no. 5, pp. 306–312, 2007.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, pp. 601–620, 2000.

[4] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, "The mutual information: Detecting and evaluating dependencies between variables," *Bioinformatics*, vol. 18, no. Suppl. 2, pp. S231–S240, 2002.

[5] M. K. S. Yeung, J. Tegnér, and J. J. Collins, "Reverse engineering gene networks using singular value decomposition and robust regression," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 6163–6168, April 2002.

[6] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, pp. 102–105, 2003.

[7] D. di Bernardo *et al.*, "Chemogenomic profiling on a genomewide scale using reverse-engineered gene networks," *Nat. Biotechnol.*, vol. 23, pp. 377–383, March 2005.

[8] F. Montefusco, C. Cosentino, and F. Amato, "CORE–Net: exploiting prior knowledge and preferential attachment to infer biological interaction networks," *IET Syst. Biol.*, vol. 4, no. 5, pp. 296–310, September 2010.

[9] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," in *Proc. Pacific Symposium on Biocomputing (PSB'99)*, Hawaii, USA, January 1999.

[10] D. di Bernardo, T. S. Gardner, and J. J. Collins, "Robust identification of large genetic networks," in *Proc. Pacific Symposium on Biocomputing (PSB'04)*, Hawaii, USA, January 2004, pp. 486–497.

[11] F. Amato, C. Cosentino, W. Curatola, and D. di Bernardo, "Identification of regulatory pathways of the cell cycle in fission yeast," in *Proc. 6th IFAC Symposium on Modelling and Control of Biomedical Systems (MCBMS'06)*, Reims, France, September 2006.

[12] F. Montefusco, C. Cosentino, J. Kim, F. Amato, and D. G. Bates, "Reverse engineering partially-known interaction networks from noisy data," in *Proc. 18th IFAC World Congress*, Milano, Italy, August 28 - September 2 2011.

[13] T. Abatzoglou, J. M. Mendel, and G. A. Harada, "The constrained total least squares technique and its application to harmonic superresolution," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1070–1087, 1991.

[14] O. Nelles, *Nonlinear System Identification*. Berlin, Germany: Springer-Verlag, 2001.

[15] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference," *Proc. Natl. Acad. Sci. USA*, vol. 107, no. 14, pp. 6286–6291, April 2010.

[16] R. J. Prill, D. Marbach, J. Saez-Rodriguez, P. K. Sorger, L. G. Alexopoulos, X. Xue, N. D. Clarke, G. Altan-Bonnet, and G. Stolovitzky, "Towards a rigorous assessment of systems biology models: the DREAM3 challenges," *PLoS ONE*, vol. 5, no. 2, p. e9202, 2010.

[17] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating realistic in silico gene networks for performance assessment of reverse engineering methods," *Journal of Computational Biology*, vol. 16, no. 2, pp. 229–239, 2009.

[18] F. d'Alché–Buc and V. Schachter, "Modeling and simulation of Biological networks," in *Proc. International Symposium on Applied Stochastic Models and Data Analysis (ASMDA'05)*, Brest, France, May 2005.

[19] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, "Gene regulatory network inference: Data integration in dynamic models – A review," *BioSystems*, vol. 96, pp. 86–103, 2009.

[20] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 22, pp. 14 031–14 036, 2002.

[21] J. Kim, D. G. Bates, I. Postlethwaite, P. Heslop-Harrison, and K. H. Cho, "Linear time-varying models can reveal non-linear interactions of biomolecular regulatory networks using multiple time-series data," *Bioinformatics*, vol. 24, pp. 1286–1292, 2008.

[22] G. H. Golub, M. Health, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979.

[23] D. L. Olson and D. Delen, Eds., *Advanced Data Mining Techniques*. Springer, 2008.

[24] K. Y. Yip, R. P. Alexander, K. K. Yan, and M. Gerstein, "Improved reconstruction of *in silico* gene regulatory networks by integrating knockout and perturbation data," *PLoS ONE*, vol. 5, no. 1, p. e8121, January 2010.